

(11) (A) No. 1059630

(45) ISSUED 790731

**BEST AVAILABLE COPY**

(52) CLASS 354-41  
C.R. CL.

(51) INT. CL. <sup>2</sup> G06K 7/00, G06F 15/30,  
H04Q 9/00

(19) (CA) **CANADIAN PATENT** (12)

(54) TRANSACTION EXECUTION SYSTEM WITH SECURE  
DATA STORAGE AND COMMUNICATIONS

(70) Anderson, Thomas G. ; Boothroyd, William A. ;  
Frey, Richard C., U. S. A.

Granted to International Business Machines  
Corporation, U. S. A.

(21) APPLICATION No. 225,121

(22) FILED 750417

(30) PRIORITY DATE U. S. A. (483, 084) 740625

No. OF CLAIMS 19

1

TRANSACTION EXECUTION SYSTEM WITH  
SECURE DATA STORAGE AND COMMUNICATIONS

Abstract of the Disclosure

10 A transaction execution system includes a host data processing system having a multiple account data base and a plurality of transaction terminals in communication with the host. The terminals each include a keyboard, a display, document handling subsystems, a hardware control subsystem, a communication subsystem and a programmable control subsystem supervising the other subsystems. A user initiates a transaction request by inserting a card into one of the terminals. After reading acceptable account identification information from the card the terminal requests entry of a pre-assigned personal ID number through the keyboard. The ID number is encrypted by the terminal at least once and communicated to the host along with information read from the card and entered via the keyboard. The host accesses from its stored data base an encrypted ID number corresponding to the received card information and makes a verification comparison of the stored encrypted ID number with the encrypted ID number received from the terminal. By requiring the entry of a nonencrypted ID number at a terminal while storing only encrypted ID numbers at the host, the correspondence between credit card account information and ID numbers need be known only to a few key personnel having access to both the encryption algorithm and a particular key therefor.

20

1059630

1                   Background of the Invention

2                   1. Field of the Invention

3                   This invention relates to transaction execution systems  
4 and more particularly to secure transaction execution systems having  
5 a central data base in communication with remote terminals which  
6 permit the execution of transactions such as the issuance of cash  
7 or the interaccount transfer of funds.

8                   2. History of the Prior Art

9                   For reasons of public convenience and economy a variety  
10 of systems have been developed for executing user requested trans-  
11 actions. One example is a check cashing machine. Such a machine  
12 reads data from a check inserted therein and issues cash equal to  
13 the amount of the check if the check is found to be in order. Other  
14 systems have been developed for use in conjunction with credit cards.

15                  One credit card system stores credit card account infor-  
16 mation in a central data base. In response to the submission of an  
17 account number from a remote terminal, the system provides infor-  
18 mation relating to the account. For instance, the system may  
19 indicate that the card has expired, that it has been stolen or may  
20 indicate the dollar amount of available credit. After a trans-  
21 action is completed the system properly adjusts the stored infor-  
22 mation to account for the transaction.

23                  Other credit card systems, which are frequently used by  
24 banks to extend their services during times of heavy business or  
25 business closure, permit the issuance of cash or the receipt of  
26 deposits through a terminal. Such a terminal typically includes  
27 a mechanism for receiving and reading information from a credit  
28 card, a keyboard, a display and document entry and exit apertures.  
29 The terminal may operate in conjunction with a data base or as a  
30 stand alone unit. Increased security for the issuance of cash



1 without human intervention is attained by issuing a personal ID number  
with each credit card. A credit card transaction is then enabled only  
when an ID number corresponding to the account number read from the  
credit card is entered through the keyboard. This required correspon-  
dence prevents a thief or mere finder of a credit card from receiving  
cash from a terminal. If a terminal operates in conjunction with a  
data base the correspondence between account numbers and ID numbers  
can be chosen at random, but frequently the ID number is derivable  
from the account number in accordance with a predetermined code. This  
10 predetermined relationship permits a stand alone terminal to check the  
ID number by algorithmically relating the ID number to the account  
number.

While this dual credit card and ID number identification technique  
improves the security of cash issue terminals, there are still weak-  
nesses that may be exploited to gain access to the large amounts of  
cash that are stored in the terminals. For instance it may be neces-  
sary to employ a substantial number of computer operators, programmers,  
analysts and other people at the host data base who have at least  
limited access to information stored in the host data base. It would  
20 be possible for any of these people to compile lists of account num-  
bers and corresponding ID numbers to be used in conjunction with  
forged or stolen credit cards to obtain cash.

An equally serious problem relates to the security of the en-  
cryption algorithm for terminals which are capable of stand alone  
operation. A large number of operators or maintenance personnel are  
required for the day-to-day support of cash issue terminals. For  
example, one or two people at each branch bank location may have  
internal access to the cash issue terminals. Often times these people  
may have access to the encryption key for normal maintenance. Alter-  
30 natively, with only a little training these people could learn

1059630

1 to acquire the key by measuring electrical signals on the internal circuitry. Once the encryption key is acquired, a correspondence between a large number of account numbers and ID numbers could be generated.

Another possible security problem arises from the transmission of account information and ID information between a terminal and a host data base. These transmissions often involve utility communication lines and are therefore subject to monitoring by a large number of people. Encryption is often used to improve communication security but anyone who is able to break the code or gain access to the code would be able to extract and compile a list of correspondence between credit card account information and ID numbers by monitoring these transmissions. In addition, by generating fake terminal communication traffic a person might gain access to the host data base and fraudulently transfer funds within data base accounts. Thus, while protected against a common thief, conventional systems which use this dual identification technique are not adequately protected against a sophisticated thief having knowledge of modern data processing equipment.

20 Summary of the Invention

A transaction execution system in accordance with the invention includes a host data processing system having a data base of stored information for many accounts and a plurality of transaction terminals. The host operates to approve or disapprove indicated transactions, modify stored account information to properly account for executed transactions and provide support information for the terminals. The transaction terminals are operatively stand alone units which are connected for communication with the host from scattered locations. Each terminal includes a document handling subsystem for cash or transaction statements, a credit card reading

1 subsystem, a host communication subsystem, a user communication subsystem, and an operational control subsystem including a programmable microprocessor.

The document handling subsystem includes a cash storage mechanism, a transport mechanism for issuing cash to a user under the supervision and control of the microprocessor and a transaction statement dispenser issuing printed statements under control of the microprocessor. The credit card reading subsystem operates under control of the microprocessor to receive and read user credit cards which may be either returned or retained after the processing of a transaction request. The host communication subsystem provides an interface for the proper transmission of information between a terminal and a host in accordance with predetermined communication formats. The user communication subsystem operates in response to the microprocessor to control user access to the terminal and includes a keyboard receiving user commands and a display interactively providing user guidance.

20 A user wishing to execute a transaction must insert a credit card into a terminal and then enter personal ID and transaction request information through the keyboard. The terminal then optionally encodes a selected portion of the credit card information using a first encryption key to obtain encrypted ID information which may be tested for correspondence with a selected portion of the keyboard entered ID information. In the absence of a predetermined correspondence the transaction is terminated, the host is informed via a message and the host reply defines the action against the card which is selectively returned or retained. If correspondence is found, the entered ID information is encoded using a second encryption key which may be the same as the first encryption key. The encrypted ID information is combined with variable information

1059630

1 such as a sequential transaction number or cash count to prevent  
2 repetitive transmission of identical encryption fields and then  
3 encoded again using a third transmission key. This encryption  
4 process allows the host data base to have stored not the ID  
5 number, but only an encrypted ID number. The data base is thus  
6 secured against the surreptitious extraction of an account number  
7 and ID number correspondence list from which counterfeit cards  
8 could be created. The encrypted ID information is combined with  
9 clear text request and credit card information and is then com-  
10 municated to the host data processing system. A three part trans-  
11 action execution sequence begins with a transaction request message  
12 which provides the host with the encrypted ID number, which is com-  
13 bined with variable data and reencrypted, credit card information  
14 and transaction request information entered through the keyboard.  
15 For example, the user might request the issuance of \$100 from his  
16 credit card account. Upon receipt of a request the host checks for  
17 correspondence between the transmitted encoded ID number and the  
18 encoded ID number stored in its data base, checks for account re-  
19 strictions such as a maximum credit limit, and if everything is in  
20 order transmits a reply message authorizing the transaction. If all  
21 is not in order the host disapproves the requested transaction.

22 Like the request message, the subsequent reply message  
23 includes an encrypted portion containing an action command and  
24 variable data such as a cash count number or a transaction number.  
25 After the encoded information is combined with clear text infor-  
26 mation such as transaction statement information and display in-  
27 formation the reply message is sent to the requesting terminal.

28 Upon receipt by the requesting transaction terminal of  
29 the transaction reply message, the terminal decrypts, checks the  
30 accuracy of the variable data to insure against error and then

1059630

1 executes the commanded actions. The terminal then generates a  
2 status message to inform the host of the execution or cancellation  
3 of the transaction and of any error conditions at the terminal.  
4 An encrypted portion of the status message includes the transaction  
5 number, the number of status bytes in the message and the cash  
6 counter status. The host responds by properly accounting for the  
7 indicated transaction by recording the transaction or updating the  
8 data base. If an error condition is indicated the host may transmit  
9 a command message to attempt to correct the error or close the  
10 terminal if the error cannot be corrected. Use of this data message  
11 technique makes the encryption keys very difficult to break and  
12 provides a communication redundancy to insure that the host and a  
13 terminal are responding to correct messages. In addition, the  
14 correspondence between personal ID numbers and account numbers is  
15 protected by an encryption scheme that avoids the necessity of  
16 storing both in the host data base.

17 Brief Description of the Drawings

18 A better understanding of the invention may be had from  
19 a consideration of the following detailed description taken in  
20 conjunction with the accompanying drawings in which:

21 Fig. 1 is a functional block diagram representation of  
22 a transaction execution system in accordance with the invention;

23 Fig. 2 is a functional block diagram representation of  
24 a transaction terminal used in the transaction execution system  
25 shown in Fig. 1;

26 Fig. 3 is an operational block diagram representation of  
27 the manner in which a user initiated transaction request is initially  
28 processed by a transaction terminal;

29 Fig. 4 is an operational block diagram representation of  
30 the manner in which transaction requests received by a transaction



## 1059630

1 terminal are processed by a host data processing system; and  
2 Fig. 5 is an operational block diagram representation  
3 of the manner in which a transaction reply message from a host  
4 is processed by a transaction terminal.

### 5 Detailed Description

#### 6 Table of Contents

7	Transaction Execution Terminal . . . . .	12
8	Terminal Information Bus . . . . .	12
9	Processor Support Subsystem. . . . .	13
10	Mechanical Control Subsystem . . . . .	14
11	User Communication Subsystem . . . . .	16
12	Transaction Statement Dispenser Subsystem. . . . .	17
13	Operator Function Subsystem. . . . .	18
14	Communication Subsystem. . . . .	19
15	Remote Connector . . . . .	19
16	Communication Message Format . . . . .	19
17	Transaction Message Assembly . . . . .	46
18	1. Transaction Request Message . . . . .	47
19	2. Transaction Reply Message . . . . .	56
20	3. Execution and Status Message. . . . .	60

**1059630**  
INTRODUCTION

1 A transaction execution system 10 in accordance with the invention includes a host data processing system 12 and a plurality of user transaction terminals 14 in communication therewith. The host data processing system 12 includes a host central processing unit 16 such as an IBM system 370, a communication controller 18 such as an IBM 3705 and a data base 20 which may include electrically alterable random access memory, magnetic tape transports, and magnetic disks. The host CPU performs the arithmetic and logical operations which are required for controlling the operation of the host data processing system 12 and processing information which is received through the communication controller 18 or stored in the data base 20. The data base 20 stores information which is related to each customer of the host central processing system 12. For instance, for a banking customer, the data base might store account information for credit card, savings, checking or other accounts of the bank as well as payroll information and information relating to the financial status of the bank's operations. Each account might be typically addressable in accordance with an account number and have stored therein the current account information such as the current balance, a history of account transactions for a predetermined period of time, encoded personal ID numbers for persons who are authorized to use the account, a maximum credit limit, and any other information the bank may wish to store as part of an account. The communication controller 18 acts as an interface between the CPU 16 and a plurality of communication channels 20. The controller 18 arranges information received by the host 16 into a communication discipline and maintains communication synchronization.

30 A transaction terminal 14 may be connected for communication with the host data processing system 12 in an almost unlimited

1 number of ways with the various methods shown in Fig. 1 being only  
exemplary. For instance, a terminal may be connected directly to  
the communication controller 18 by either a local communication  
link such as cable 24 for local user transaction terminal 26 or a  
utility or radio link 28 for a remote user transaction terminal 30.  
Alternatively, a terminal may be connected to the host central  
processing system 12 through a controller 32 such as an IBM 3601  
by either direct connection to the controller 32 as by cable 34 for  
terminal 36 or by connection in a communication loop 38. Although  
10 other devices may be included, the communication loop 38 is il-  
lustrated by way of an example as including a first teller work  
station 40, a second teller work station 42, a first user trans-  
action terminal 44 and a second user transaction terminal 46. While  
the communication loop 38 may include remote transmission links  
such as radio communication or communication over commercial  
utility lines, for a bank system, the controller 32 might typically  
be located at a branch bank with all the controller 32 might typically  
be located at a branch bank with all data processing terminals at  
the branch being connected into the loop 38. The controller 32  
20 may itself be connected to a communication channel 22 of communica-  
tion controller 18 either directly through a communication link 48  
such as a utility communication line as shown in Fig. 1 or may  
itself be connected in a communication loop such as the loop 38  
which extends to a communication channel 22 of communication con-  
troller 18. A description of one such communication system may be found  
in United States Patent No. 3,921,137, issued November 18, 1975  
entitled "SEMI STATIC TIME DIVISION MULTIPLEX SLOT ASSIGNMENT" by  
C. McClearn and T.A.C. Miller.

1 In general, the controller 32 merely acts as a relay device for information which is passed around the loop 38 but may also serve as the host data processing system when immediate, real time communications with the host data processing system 12 are not maintained. When serving as the host, the controller 32 must store transaction execution information for later processing by the system 12 and must provide host support functions which are required for operation of a terminal 14.

#### TRANSACTION EXECUTION TERMINAL

10 While the particular manner in which a transaction terminal 14 is implemented is not critical to the practice of this invention, a preferred embodiment of the transaction terminal 14 is shown in Fig. 2. The terminal 14 is generally modular in nature and includes a programmable microprocessor 60 coupled to a plurality of terminal subsystems by an information bus 62. The microprocessor 60 is driven by a clock signal from clock signal generator 64 and is operationally connected to a data storage module 66 providing both electrically alterable random access memory (RAM) and read only storage (ROS). The read only storage portion of the data storage  
20 66 stores the various operating programs for the microprocessor 60. The random access memory portion of data storage module 66 provides a scratchpad for program execution. With typical IC memories the contents of the RAM are lost in the event of a power failure.

#### TERMINAL INFORMATION BUS

The microprocessor 60 communicates with the modular sub-  
systems solely through the terminal information bus 62. This technique of interconnecting modular subsystems with the microprocessor 60 through the bus 62 permits the microprocessor 60 to receive detailed information on the terminal status and maintain detailed  
30 direction of terminal hardware operations without a large number

## 1059630

1 of input and output information connections. The task of sensing terminal status information is performed by the individual terminal subsystems. This information is then transferred to the microprocessor 60 on command from the microprocessor 60. Similarly, the driver circuitry and hardware for executing microprocessor commands is contained within the subsystem modules. The microprocessor commands are extremely basic and detailed in nature. Each command accomplishes a basic subsystem operation such as the activation or deactivation of a motor, the display or printing of a character, the feeding of a bill or the reading of a communication character. 10 The information bus 62 includes a system reset signal, 9 data input signals (8 bits + parity) for carrying information to the processor 60, 9 data output signals (8 bits + parity) for carrying information from the microprocessor 60 to an operably connected subsystem, and bus control signals for controlling the transfer of information onto and off from the bus 62.

### PROCESSOR SUPPORT SUBSYSTEM

One of the operational subsystems which is connected through bus 62 to micro-processor 60 is processor support subsystem 68. 20 Processor support subsystem 68 provides hardware assistance to the microprocessor 60 in contrast to other terminal subsystems which have functions related to particular aspects of terminal 14 operation.

Processor support subsystem 68 receives a 1 MHz clock signal from clock signal generator 64 and divides this signal to generate lower frequency clock signals which are used in the other subsystems. One lower frequency clock signal is utilized for the generation of period interrupt commands at 10 msec. intervals. These interrupt commands cause interrupt logic within processor support 30 subsystem 68 to generate a microprocessor interrupt every

1 10 msec. The microprocessor 60 utilizes these clock period inter-  
2 rupts to maintain an event control time base for the various  
3 operations of the terminal 14. Reset logic within subsystem 68  
4 controls the reset line of the information bus 62. Activation  
5 of this reset line causes initialization of the processor 60 as  
6 well as all modules which are connected to bus 62 and cancels  
7 any pending user transaction. The processor 60 is returned to  
8 a predetermined program instruction from which program execution  
9 can begin anew following the reset. The reset signal is activated  
10 in response to AC power on, a reset switch, or a hang signal from  
11 a hang detector within operational hardware subsystem 68. The  
12 hang detector monitors the control lines of the bus 60 and generates  
13 a hang signal when bus activity ceases for a length of time which  
14 is sufficient to indicate that the microprocessor 60 is not operating  
15 properly. A run detector responds to the timer interrupt request  
16 signals and generates a run signal which is maintained active so  
17 long as the microprocessor regularly responds to the requests. If  
18 a predetermined period of time elapses without the processing of  
19 a timer interrupt request, the run detector terminates the run  
20 signal. The processor support subsystem 68 also includes read  
21 data logic which receives a string of serial information as it is  
22 read from a user credit card, separates the data from the clocking  
23 information, deserializes the binary bit stream and places the  
24 information on the bus 62 for processing by the microprocessor 60.

#### 25 MECHANICAL CONTROL SUBSYSTEM

26 A mechanical control subsystem 70 provides the actual  
27 mechanical manipulation of various hardware features of the terminal  
28 14. Subsystem 70, which like the other subsystems has no branching  
29 or decision making capability, executes basic, elemental commands  
30 from the microprocessor 60 and collects information on the physical

1 status of the various hardware functions for communication back to  
2 microprocessor 60. As an example of the individual elementary  
3 nature of functions which are executed by mechanical control sub-  
4 system 70, a credit card handler mechanism responds to a credit  
5 card direction and move commands to activate a motor which drives  
6 a card conveyor system to move the credit card beneath a read head.  
7 Sensors (switches or photocells) are positioned to sense the pre-  
8 sence of the credit card at (1) entry, (2) exit jam sensor, and (3)  
9 card escrow positions. When a sensor is activated an information  
10 bit is available in a status word to indicate this condition. When  
11 the microprocessor 60 periodically reads the various status words  
12 during a read operation it determines that the credit card has  
13 reached the escrow area where the card is held. Processor 60 then  
14 commands that the credit card feed motor be reversed for a short  
15 period of time to "brake", and then commands that the motor be turned  
16 off. In similar elemental fashion, the mechanical control subsystem  
17 70 controls the complete processing the credit card such as retention  
18 or return to the user. Other functions include control of the  
19 depository wherein the user may deposit documents which are passed  
20 into a retention bin in such a manner that the user never has  
21 access to the retention bin. Similarly, the mechanical subsystem  
22 70 controls the opening and closing of user access doors and the  
23 issuance of predetermined amounts of cash to an escrow area at  
24 which printed transaction statements may also be accumulated along  
25 with the cash and the issuance or retention of documents presented  
26 to the escrow area. In addition to sensing the status of mechanical  
27 hardware which is manipulated by mechanical control system 70, the  
28 control subsystem 70 senses the presence of cash stored by the cash  
29 issue hardware and indicates when there is not enough cash available  
30 to execute a maximum issue transaction. Subsystem 70 also senses

1 several conditions that may be communicated to a remote control panel  
as well as the processor 60. These remote signals include an indi-  
cation of whether the service door is opened, whether or not a pene-  
tration sensing grid has been disturbed, and whether or not an  
"intervention required" condition exists. Other signals which may  
be communicated to the remote panel include transaction statement  
forms or cash low, operator access service door open, communication  
between the terminal and host ready. Command switches located on  
a remote panel may include a terminal reset switch and a wrap switch  
10 which commands a test of the communication link.

#### USER COMMUNICATION SUBSYSTEM

A user communication subsystem 72 controls bidirectional com-  
munications between the terminal 14 and a user. The communication  
subsystem 72 includes a keyboard for receiving user generated com-  
mands, a display of 222 horizontal dots by 7 dots and includes dis-  
play control logic and a refresh buffer. The display control logic  
receives the "dot image" of the particular display and then con-  
tinues the display until a contrary command is received.

20 The keyboard is divided into several fields with a plurality  
of keys in each field. For instance, a transaction selection field  
indicates the type of transaction a user wishes to execute. Other  
fields include a from account select field indicating an account  
from which funds are to be taken, a to account select field indi-  
cating an account to which funds are to be deposited and a numeric  
keyboard field permitting the entry of decimal numbers such as  
personal ID numbers or dollar amounts. "Back lights" are provided  
on the function select, to account, and from account keys to generate  
an audit trail indicating to a user which keys have been selected in  
previously used fields. All back lights are illuminated in the  
30 field in which the next key activation should



1 occur. For instance, as a user inserts his credit card into the terminal 14 he is requested to key in his personal ID number. After proper receipt of the ID number all of the keys in the function selection field would become lighted. As the user activates a particular key, such as a funds transfer key, the other back lights are extinguished with only the funds transfer key remaining backlighted. All keys in the next field such as the from account field are then illuminated in preparation for the next step in the transaction request. In this way an audit trail is provided to indicate previous selections and the next selection field is also indicated. Display messages and color coding may also be used to guide the user in the proper sequence. The keyboard control logic of the user communication subsystem 72 includes the circuitry necessary to back light specific keys commanded by the microprocessor 60 and to indicate to the microprocessor which keys have been activated by a user.

#### TRANSACTION STATEMENT DISPENSER

20 A transaction statement dispenser subsystem 74 includes a form handler for transporting transaction statement forms, a printer, printer control logic and logic for interfacing the subsystem 74 with bus 62. The transaction statement dispenser subsystem 74 performs only specific, basic commands such as starting movement or printing of specific characters. The subsystem 74 collects information on the physical status of the transaction statement dispenser hardware for communication through bus 62 to the microprocessor 60. This information is then used by the microprocessor 60 which operates under program control to detect the successful completion of a particular elemental function and commands the initiation of additional functions.

OPERATOR FUNCTION SUBSYSTEM

1  
2           An operator function subsystem 76 provides operator  
3 maintenance interfacing and includes entry switches, a four digit  
4 hexadecimal display, power sense circuitry, a 128 byte power off  
5 protected auxiliary memory which is used for storing system  
6 parameters and logging exception information. Stored parameters  
7 include a cash counter number, encryption keys and a transaction  
8 number. Access to the operator panel is through a double locking  
9 door at the rear of the terminal 14 which must be closed for  
10 user operation. Opening of the access door and attempting a  
11 maintenance function causes destruction of encryption keys which  
12 are normally stored in this auxiliary memory. This destruction of  
13 the keys provides security of the keys from an operator who might  
14 seek to use electronic instruments to read the key from the non-  
15 volatile memory. The keys must then be re-entered through the  
16 keyboard by a person of high trust before the terminal can be  
17 reopened. The 8 byte keys are each entered as 16 hexadecimal  
18 digits two digits at a time. Only the two preceding digits are  
19 displayed as the keys are entered to increase the difficulty of  
20 an interloper discovering the keys. Alternatively, a Key A which  
21 defines the correspondence between account numbers and personal  
22 ID numbers may be even further protected by requiring entry of  
23 a deencrypted Key A (Key A') which is encrypted in accordance  
24 with a fourth encryption key to produce the actual Key A. Using  
25 this technique the actual Key A can remain secure from all personnel  
26 at the physical location of the terminal 14. The power sense cir-  
27 cuitry monitors both the AC utility voltage level and the internal  
28 DC power levels and in the event of an indication that AC power  
29 is lost and the DC voltage levels are low but still usable, a  
30 signal is sent to the microprocessor 60 causing critical information

1059630

1 to be saved and then access to the auxiliary memory is restricted  
2 while the memory is driven from an auxiliary power source. An  
3 indication signal is provided to the operator panel so long as the  
4 DC logic voltages are adequate.

5 COMMUNICATION SUBSYSTEM

6 A communication subsystem 78 provides communication  
7 interfacing between a communication channel and the information bus  
8 62. Communication subsystem 78 is conventional in nature and  
9 receives information from or provides information to terminal  
10 information bus 62 one byte at a time.

11 REMOTE CONNECTOR

12 A remote signal connector 82 permits the connection of  
13 some status signals and some control signal inputs to a remote  
14 control panel which is actually part of the terminal 14. For  
15 instance, a bank branch might have five terminals 14 and a single  
16 centralized remote control panel with optical displays and control  
17 switches for each of the five terminals 14 at a convenient central-  
18 ized location. These remote signals are primarily for monitoring  
19 terminal operation or controlling special conditions and are not  
20 utilized for normal user transaction. The particular remote panel  
21 has been previously explained.

22 COMMUNICATION MESSAGE FORMAT

23 There are essentially two different types of messages which  
24 may be sent from a terminal 14 to a host data processing system  
25 and four types of messages which may be sent from the data processing  
26 system 12 to a host transaction terminal 14. The terminal to host  
27 messages include a transaction request message which is the normal  
28 first communication message following a user initiated transaction  
29 and a status message which is typically the last of a three message  
30 sequence. There are two basic types of status messages. The first

1 is a reply status message which serves as the third communication  
2 message in a normal user transaction sequence and informs the host  
3 of the completion or cancellation of a user requested transaction.  
4 The second is an exception status message which indicates a status  
5 or condition for a terminal 14 other than a normal operating con-  
6 dition. For example, an exception status message would be sent  
7 in reply to an inquiry command from the host data processing system,  
8 when the service door is opened, upon detection of a serious error  
9 condition such as a user door jam or a hard machine failure or any  
10 time initialization is required.

11           The four types of messages which may be transmitted from  
12 a host data processing system 12 to a transaction terminal 14  
13 include a transaction reply message, command message, a load init-  
14 ialization message, and an echo message. The transaction reply  
15 message is the normal response to a transaction request message  
16 during the course of a normal user transaction and informs the  
17 terminal 14 of the manner in which the requested transaction should  
18 be completed. A command message commands changes in a terminal  
19 14 logical state and may also serve as an inquiry for a status  
20 message if no changes are desired. A load initialization message  
21 is sent from a host to a terminal 14 in response to an exception  
22 status message requesting initialization (IPL). The load initiali-  
23 zation message contains message text, option selection information,  
24 font tables, program routines, and data information for storage  
25 in the volatile random access portion of data storage 66 of  
26 the microprocessor 60 within a terminal 14. An echo message is  
27 used as a diagnostic assurance test and can be sent only when a  
28 terminal 14 is in a closed state. The terminal 14 responds to  
29 an echo message with an echo message.

1           There are only three basic message sequences which may be used for the communication of messages between a terminal 14 and a host data processing system 12. A single message sequence consists of an exception status message transmitted from a terminal 14 to a data processing system 12. The exception status message may either indicate that an abnormal condition has occurred or be a request for initialization. A "command message" from the host is not required. The message contents indicate which is the case.

10           A two message sequence may include either a command message from host, a load initialization message, processing system 12 to a terminal 14 followed by an appropriate status message from the terminal 14 to the host data processing system 12 or a host echo message followed by a terminal echo message. The transaction terminal 14 will reject a command that is received while the terminal is processing a previous command, an unintelligible message, or an unrequested transaction reply message. In each instance the host may be either a remote system or a directly connected local system.

20           Each time the terminal 14 assumes an initial power on condition, for whatever reason, the terminal 14 must request and receive a load initialization message from the host data processing system before the terminal 14 can be reopened to accept transactions. Transaction terminals such as terminals 36, 44, and 46 in Fig. 1, which are connected to a controller 32 may operate in an off-line mode. Under such circumstances, the controller 32 serves as the host data processing system and merely records user transactions, for example on magnetic tape or disc. The transaction information is then made available to a transaction accounting system at a later time to permit the updating of accounts. If operating on-line mode, some host functions may be handled by the controller 32

## 1059630

1 such as storage of the initialization program for the terminals,  
2 but normally all communications are merely communicated to the  
3 host data processing system 12 without change. In such an on-line  
4 mode of operation, the host data processing system 12 may update  
5 account records stored in its data base in real time; that is as  
6 user requested transactions are executed.

7       Each time a terminal 14 loses power information is lost  
8 from the RAM portion of data storage 66, and initialization must  
9 be requested at power turn-on. After the receipt of initialization  
10 information from the host a terminal 14 may be opened to receive  
11 user transactions, but only on command from the host. Initialization  
12 is accomplished by a terminal 14 using the single message format  
13 to send an exception status message requesting initialization.  
14 The host data processing system then initiates a new communication  
15 sequence by sending an initialization message (in multiple parts)  
16 containing the requested initialization information. Upon success-  
17 fully receiving the initialization information the requesting  
18 terminal 14 completes the two part message sequence by sending a  
19 status message back to the host data processing system.

20       Every message which is sent between a transaction terminal  
21 14 and a host data processing system 12 begins with a four byte  
22 header field. Byte 1 of the header field is a message length  
23 byte (L) containing a binary count of the number of message bytes  
24 in the message text (including L). Byte 2 is a 1 byte transaction  
25 sequence number (N) in binary form. This number is incremented  
26 for each new user transaction and is included in all messages  
27 exchanged for that transaction. The number has a range of 1 to  
28 255 inclusive. Zero (hex 00) is used for messages that do not  
29 relate to a user transaction. Thus, a transaction number counter  
30 which is incremented for each new user transaction overflows from

## 1059630

1 hex FF to hex 01. The transaction number (N) is stored in the  
2 power out protected auxiliary memory of operator function sub-  
3 system 76 so that it remains available after a short term power  
4 outage. Byte 3 of the common header field is a class byte (C)  
5 which identifies the type of message and thus the format of the  
6 message which is being sent. Byte 4 is the final byte of the header  
7 field and identifies a message sub-class (SC) which serves as a  
8 modifier to the message class byte.

9       Only a few of the possible combinations of message  
10 classes (C) and sub-classes (SC) are actually implemented.  
11 Class hex 01 identifies a transaction request message from a  
12 terminal 14 to a host data processing system. Within class 01,  
13 nine sub-classes have been implemented. Sub-class hex 00 indicates  
14 that a user requested transaction is incomplete because the ID  
15 number has not been properly entered. Sub-class hex 01 indicates  
16 a cash issue request. Sub-class hex 02 indicates an account inquiry.  
17 Sub-class hex 03 indicates that a user is requesting to deposit  
18 funds. Sub-class hex 04 indicates that a user is requesting to  
19 transfer funds from one account to another. Sub-class hex 05  
20 indicates that a user is requesting to pay a loan or bill by  
21 depositing money in the transaction terminal. Sub-class hex 06  
22 indicates a special transaction wherein the nature of the trans-  
23 action is identified by entry of a predetermined number through  
24 the keyboard rather than by activation of a single key in the  
25 transaction selection field of the keyboard. Sub-class hex 07  
26 indicates that a requested transaction is incomplete because the  
27 deposit flap covering the deposit bin has been jimmied. Sub-class  
28 hex 08 indicates a user request to pay a bill or loan by the trans-  
29 fer of funds from one account to another.

## 1059630

1           A class of message designated C = hex 15 identifies a  
2 status message from a terminal 14 to a host data processing system  
3 12. There are five sub-classes of messages under this class.  
4 Sub-class hex 01 indicates a transaction completion status message.  
5 Sub-class hex 02 indicates that the message is in response to the  
6 execution of a command and the status number, N, in the common  
7 header must be set to 0. Sub-class hex 03 is an exception status  
8 message indicating an error condition or requesting initialization  
9 and the transaction number N must be set to 0. Sub-class hex 04  
10 indicates that the status message is in response to initialization  
11 and the transaction number N must be set to 0. Sub-class hex 08  
12 is a recovery request or command response message and the trans-  
13 action number N must be set to 0 for this message. A recovery  
14 request indicates that the host has lost track of the current trans-  
15 action and requires an update. The terminal responds with an  
16 exception status message.

17           A transaction reply message from a host data processing  
18 system to a transaction terminal 14 is indicated by class hex 0B.  
19 There are nine sub-classes indicated by the sub-class byte under  
20 this sub-class. Sub-class hex 00 indicates that the transaction  
21 is incomplete because the ID number was not properly entered. Sub-  
22 class hex 01 indicates a cash issue transaction request. Sub-class  
23 hex 02 indicates an account inquiry transaction request. Sub-class  
24 hex 03 indicates a deposit transaction request. Sub-class hex 04  
25 indicates a funds transfer transaction request in which funds are  
26 to be transferred from one account to another. Sub-class hex 05  
27 indicates a transaction request for the payment of a loan or bill  
28 by transfer of funds deposited in the terminal to an account. Sub-  
29 class hex 06 indicates a special optional selection transaction  
30 wherein the nature of the transaction is determined in accordance



# 1059630

1 with a number entered through the numerical keyboard rather than  
2 by the activation of a single key in the transaction selection  
3 field of the user keyboard. Sub-class hex 07 indicates that the  
4 message relates to a requested user transaction which is incomplete  
5 because the deposit flap of the terminal 14 has been jimmied. Sub-  
6 class hex 08 indicates a user transaction wherein a loan or bill  
7 is to be paid by transferring funds from one account to another.

8 Class hex 0C identifies a command message from the host  
9 data processing system to a terminal 14. A command message does  
10 not relate to a particular transaction and therefore the transaction  
11 number N of the header field is always set to 0. Sub-class hex 01  
12 indicates an open command. Sub-class hex 02 indicates a command  
13 to close the transaction terminal 14. Sub-class hex 03 indicates  
14 an inquiry type of message in which a transaction terminal 14 may  
15 not perform any function in response to the command but must  
16 respond with a status message. Sub-class hex 04 indicates a command  
17 to change the third key (key B) which is the transmission encryption  
18 key from the present key to a key contained within the message. Sub-  
19 class 05 indicates a command to set the transmission encryption key  
20 (key B) using a back up key (key C). Sub-class hex 06 indicates  
21 that a transaction terminal 14 is commanded to request an initial  
22 program load. Sub-class hex 07 indicates that the message includes  
23 a command to either change the optical display or contains a  
24 written message to be printed by the transaction statement dispenser.  
25 Sub-class hex 08 is a command for transaction terminal 14 to send a  
26 class hex 15 sub-class hex 08 recovery request message back to the  
27 host.

28 The load initial program message from the host to a  
29 transaction terminal is designated class hex 0D and has only one  
30 sub-class which is designated hex 01.

## 1059630

1           An echo message from the host data processing system  
2 to a terminal 14 is designated by class hex 10. Within this class  
3 there are four sub-classes of echo messages. Sub-class hex 00 is  
4 the basic echo message and merely commands the transaction terminal  
5 14 to retransmit the echo message back to the host data processing  
6 system. Sub-class hex 01 indicates an echo canned message command  
7 which is both checked for bit pattern and echoed. The bytes of  
8 data in the canned text are designed to send all possible bit  
9 patterns to check the operation of communication facilities.  
10 The message pattern is retained by the terminal for comparison  
11 with a second transmission of the message pattern. An echo variable  
12 record sub-class designated hex 02 is similar to the canned echo  
13 sub-class except that the message may contain host entered data.  
14 The transaction terminal echos the message back and also retains  
15 the message in storage for comparison with a second transmission  
16 of the same message. Upon receipt of the second transmission  
17 of the message, the transaction terminal checks and echoes as  
18 for sub-class 01. A log data request message is designated sub-  
19 class 03. This will cause the terminal to send the 8 most current  
20 error log records. No encryption or decryption is involved in the  
21 transmission of any echo message.

22           The four byte common header field of each message is  
23 followed by the message data in a format that depends upon the  
24 particular type of message that is being sent. For a transaction  
25 request message from the terminal 14 to the host data processing  
26 system bytes 1-4 of the common header are followed by bytes 5-8  
27 which contain a 32 bit encrypted field. This 32 bit encrypted  
28 field will be discussed in greater detail later, but in general  
29 the field includes an encrypted form of the personal ID number  
30 which was entered through the user keyboard and one byte of

## 1059630

1 varying information which may be either the contents of a cash  
2 counter or a transaction number counter.

3 Byte 9 is a from account select (FAS) byte indicating  
4 which key within from account selection field of the user key-  
5 board was activated. The data content of this ninth byte in-  
6 dicates the type of account from which the funds for the user  
7 requested transaction are to be taken. Hex 21 indicates from  
8 a checking account, hex 22 indicates from a savings account,  
9 hex 23 indicates from a credit card account, and hex 24 in-  
10 dicates from a special optional selection account, which is  
11 further defined by a numeric modifier. By making special  
12 arrangements with the bank, a user can open multiple accounts.  
13 These accounts can then be assigned predetermined three digit  
14 (decimal) numbers. By activating the special optional selection  
15 from account key on the keyboard the user is then permitted to  
16 enter up to three decimal numbers through the numerical key-  
17 board to indicate which of possibly many predefined accounts he  
18 wants debited. This account identification number is transmitted  
19 one digit per byte and bytes 10-A where A may assume the values  
20 10, 11 or 12 depending on whether the special keyboard determined  
21 account number contains 1, 2 or 3 digits respectively. Because  
22 the FAS field may have a variable length, it must be followed by  
23 a field separator (FS) byte having the data content hex FE. which  
24 is used to define the limits of variable length fields. Adjacent  
25 field separators indicate a zero length or no entry field between  
26 them. The FS byte delimits the end of the field preceding the FS  
27 byte.

28 Following the FS byte for the from account select (FAS)  
29 field is a to account select (TAS) field designating an activated  
30 key within the to account select field of the user keyboard. Hex  
31 31 indicates that funds are to be deposited to a checking account,

## 1059630

1 hex 32 indicates to a savings account, hex 33 indicates to a credit  
2 card account, and hex 34 indicates to a special optional selection  
3 to account select key which may be modified by up to three digits  
4 (decimal) immediately following the first TAS byte. These numeric  
5 modifiers have the same meaning in the TAS field as in the FAS field.  
6 Because the TAS field is variable in length it must also be followed  
7 by a field separator (FS) byte having data content hex FE. Following  
8 the field separator byte for the to account select field, the data  
9 which is read from the magnetic stripe on the credit card is trans-  
10 mitted. By removing the parity bit from the standardized code of  
11 the American Bankers Association, it is possible to pack the two  
12 four bit characters of credit card data in each byte of the message.  
13 In the event that an odd number of credit card characters appears  
14 on the credit card, the last byte is padded with a hex F to fill  
15 all bytes of the message. Start of card characters, end of card  
16 characters and longitudinal redundancy check (LRC) characters are  
17 excluded from the transmitted transaction request message in as  
18 much as they are checked by the terminal 14.

19           A status message from a terminal 14 to the host data  
20 processing system begins with the four byte common header field  
21 identifying the message length (L), transaction number (N), message  
22 class (C), and message sub-class (SC) for the message. In byte  
23 positions 1-4. Byte positions 5-8 contain a 32 bit encrypted  
24 field. This 32 bit field will be discussed in greater detail  
25 below but in general contains a repetition of the eight bit  
26 transaction number (N), eight bits which representing the revolving  
27 cash count for denomination two (CNTR2), eight bits indicating  
28 the number of status bytes (CB) and eight bits representing the  
29 revolving cash count for denomination one (CNTR1). The byte CB  
30 is a one byte field containing a binary count of a number of status

## 1059630

1 and inquiry data bytes which follow the encrypted portion (bytes 5-8)  
2 of the message for a normal status message. For a "request recovery  
3 message" the CB field contains the "action field" from the trans-  
4 action reply for the last transaction request message. The "action"  
5 field is an eight bit field transmitted as part of the 32 bit  
6 encrypted field of a transaction reply message. The eight bit  
7 counter portions (CNTR) of the 32 bit encrypted field indicates  
8 binary count of bills issued by the second and first cash issue  
9 mechanism. These numbers are taken from counters which are  
10 incremented for each issued bill and roll over from hex FF to  
11 hex 00. The counts are stored in the auxiliary memory of the  
12 operator function subsystem 76 so that the count is preserved  
13 during a short term power outage. Following the 32 bit encrypted  
14 field at bytes 5 to 8 is a data field. The data field includes  
15 a four byte status field in byte positions 9-12. These four bytes  
16 define the current status of a terminal 14 as discussed below.  
17 Most status messages terminate with an FS byte at byte position  
18 13. However, a status message which is sent in response to an  
19 inquiry command message contains 112 of the 128 bytes stored in  
20 the auxiliary memory of the operator function subsystem 76 which  
21 are transmitted behind the four status bytes. For this message the  
22 field CB would contain the number 116. The 16 bytes of the non-  
23 volatile memory which are not sent in response to an inquiry  
24 message contain the two eight byte encryption keys. If the  
25 status message is being resent in response to a request recovery  
26 message, the four status bytes contain the four bytes of the last  
27 transaction status message and are followed by the complete original  
28 transaction request message. This information then would allow  
29 the host to re-construct the conditions which existed prior to the  
30 event which caused the host to request recovery.

## 1059630

1           The 32 bit positions of the four status bytes at byte  
2 positions 9-12 of a status message each have a predetermined meaning.  
3 These meanings are assigned to define the physical and operating  
4 status of a terminal 14 with sufficient particularity that  
5 a host data processing system can assess and control the general  
6 operation of each terminal 14. These meanings are described in  
7 tabular form below with the number to the left indicating the  
8 status byte number ranging from 0 to 3 with status byte 0 in status  
9 message byte position 9 and status byte 3 in status message byte  
10 position 12. For each status byte there are 8 bits designated  
11 bit 0 - bit 7 with bit 0 being in the most significant bit  
12 position and bit 7 in the least significant bit position.

# 1059630

1	<u>Byte</u>	<u>Bit</u>	<u>Description</u>
2	0	0	Transaction completion status bit. This bit position
3			is set to logic 1 at the beginning of each transaction
4			to indicate that the transaction has not been com-
5			pleted because a transaction reply message is required.
6			The bit position is reset to logic 0 when a trans-
7			action has been executed as specified in a transaction
8			reply message.
9	0	1	Invalid transaction sequence number in transaction
10			reply bit. This bit position is reset to logic 0
11			each time a new transaction is started. The bit
12			position is set to logic 1 any time the transaction
13			number (N) within the common header field of a
14			message received from the host data processing system
15			is inaccurate. An exception is made for an echo
16			message which does not convey meaningful information
17			in the transaction number position of the header
18			field.
19	0	2	Invalid transaction subclass in reply message bit.
20			This bit position is reset to logic 0 each time a
21			new transaction is started and is set to logic 1
22			any time a transaction reply message is received
23			containing a different number in the fourth or
24			subclass byte of the common header field from that
25			of the transaction request message. Byte 0 bit 0
26			must be set simultaneously with this bit position.
27	0	2	Invalid transaction subclass in reply message bit.
28			This bit position is reset to logic 0 at the
29			beginning of each new user transaction and set to
30			logic 1 any time the subclass byte of a transaction

## 1059630

1           reply message does not match the subclass byte of  
2           the corresponding transaction request message. Byte  
3           0 bit 0 must be set each time this bit position is  
4           set.

5       0     3       Invalid class bit. This bit position is reset to  
6                    0 after an exception status message has been sent and  
7                    is set to logic 1 any time a message is received from  
8                    the host data processing system containing an invalid  
9                    class designation in byte 3 of the common header  
10                   field. As an example, a terminal 14 might receive  
11                   a nonrequested initialization (IPL) message or a  
12                   nonrequested transaction reply message.

13       0     4       Amount error in transaction reply message bit. This  
14                    bit position is reset to logic 0 at the beginning  
15                    of each new transaction and set to logic 1 any time  
16                    a transaction reply message is received with the  
17                    dollar amount byte within the encrypted field thereof  
18                    indicating an improper dollar amount. (AMT) Bit 0  
19                    of byte 0 must be set to logic 1 any time this bit  
20                    position is set to logic 1.

21       0     5       Unassigned.

22       0     6       Customer cancelled transaction bit. This bit  
23                    position is reset to logic 0 at the beginning of  
24                    each new transaction and set to logic 1 in the event  
25                    that a customer activates a cancelled key on the  
26                    user keyboard subsequent to the transmission of the  
27                    transaction of a transaction request message.

28       0     7       User timeout bit. This bit position is reset to  
29                    logic 0 at the beginning of each new user trans-  
30                    action and set to logic 1 any time a user consumes



## 1059630

1 more than an allotted predetermined length of time  
2 in entering a number through the user keyboard or in  
3 depositing materials through the deposit flap. Bit  
4 0 of byte 0 must be set any time this position or  
5 position 0 6 is set to logic 1.

6 1 0 Command reject bit. This bit position is reset to  
7 logic 0 after a command status message is sent.  
8 The bit position is set to logic 1 upon receipt of  
9 a command message which cannot be executed because  
10 the terminal 14 is busy at the time a command is  
11 received.

12 1 1 Invalid command bit. This bit position is reset  
13 to logic 1 upon sending a command status message.  
14 The bit position is set to logic 1 any time a  
15 command message is received with missing fields  
16 therein. For instance, a key change command which  
17 does not include the new key or a change display  
18 command without a new display field. This bit posi-  
19 tion is also set to logic 1 in response to a command  
20 message containing an invalid subclass designation  
21 in byte 4 of the common header field.

22 1 2 IPL request bit. This bit position is reset to  
23 logic 0 upon the proper receipt of a load initiali-  
24 zation message from the host data processing system  
25 and set to logic 1 each time a terminal 14 goes from  
26 a closed to an open condition, for example, upon  
27 closure of the operator/customer engineer access  
28 panel, or upon command from the host data processing  
29 system. This bit is also set to logic 1 each time  
30 a terminal 14 receives a command message commanding

# 1059630

1 the terminal to request an IPL.

2 1 3 IPL and process bit. This bit position serves as

3 a modifier bit for bit position 2 of byte 1. A

4 combination of bit 2, bit 3 equal 00 indicates that

5 the terminal is initialized. This condition can

6 occur only when the terminal is in an open state.

7 The combination of bit 2, 3 equal 10 indicates that

8 initialization has been requested but the load

9 initialization message has not been received. A

10 combination of bit 2, 3 equal 11 indicates that a

11 load initialization is in process.

12 1 4 Cash counter error bit. This bit position is reset

13 to logic 0 at the beginning of each new user

14 transaction. The bit position is set to logic 1

15 any time a transaction reply message is received

16 containing a cash counter byte (CNTR) within the

17 encrypted field thereof which does not match the

18 status of the cash counter within the terminal. The

19 cash counter is a rollover counter which is incre-

20 mented each time a new bill is issued. Byte 0,

21 bit 0 must be set to logic 1 each time this bit

22 position is set to logic 1.

23 1 5 C and CS field error bit. This bit position is reset

24 to logic 0 upon sending an exception status message.

25 It is set to logic 1 upon receipt of a command

26 message from the host data processing system con-

27 taining a class and subclass (C&SC) byte within

28 the encrypted data field that does not match the

29 class and subclass byte of the common header field.

30 This failure to match indicates a possible encryption

## 1059630

1 key synchronization error or host error. In a  
2 normal command message, the two class (C) and  
3 subclass (SC) bytes of the common header field are  
4 combined into a single class and subclass (C&SC)  
5 byte (packed by sensoring the four leading 0 bits  
6 of each byte).

7 1 6 Communications timeout on transaction, reply sequence  
8 bit. This bit position is reset to logic 0 at the  
9 beginning of each new user transaction. The bit  
10 position is set to logic 1 any time a predetermined  
11 period of time expires following the transmission  
12 of a user transaction request message without the  
13 receipt of a corresponding transaction reply message.  
14 Byte 0, bit 0 must be set to logic 1 any time this  
15 bit position is set to logic 1.

16 1 7 Unintelligible message bit. This bit position is  
17 reset to logic 0 after sending an exception status  
18 message. It is set to logic 1 to indicate an  
19 unintelligible message any time a message is received  
20 which does not correspond to the required predeter-  
21 mined message format. For example, the number of  
22 bytes may not agree with the message length (L)  
23 designation in the common header or a parity error  
24 may occur upon reading a data byte or a byte position  
25 may contain invalid data.

26 2 0 Card retained bit. This bit is reset to logic 0  
27 at the beginning of each new user transaction and  
28 is set to logic 1 any time a user requested trans-  
29 action is terminated with the terminal 14 retaining  
30 the credit card which was inserted therein by the

# 1059630

1			user. This bit position indicates that the card
2			was retained as a result of a hardware error at the
3			terminal 14 rather than in response to a command
4			from the host data processing system.
5	2	1	Dispense error bit. This bit position is reset to
6			logic 0 at the beginning of each new user trans-
7			action. The bit position is set to logic 1 any time
8			an error occurs during the dispensing of a document
9			such as a bill or a transaction statement. This
10			bit position is set any time a document is dumped
11			from an escrow area into a retention bin. Since the
12			transaction may be completed upon retry, this bit
13			position does not necessarily indicate an incomplete
14			user transaction.
15	2	2	Unrecoverable depository error bit. This bit position
16			is reset to logic 0 at the beginning of each new
17			user transaction. This bit position is set to logic
18			1 any time an error condition such as a jam occurs
19			in the terminal depository and the terminal is
20			unable to recover from the error condition.
21	2	3	Display table overflow bit. This bit position is
22			reset to logic 0 upon sending a status message.
23			The bit position is set to logic 1 upon receipt of
24			a change display command message from the host
25			data processing system containing more display data
26			than the terminal display system can handle. An
27			improper display message is not accepted by a
28			terminal 14.
29	2	4	Unassigned.
30	2	5	Unassigned.

# 1059630

1	2	6	Unassigned. Intervention required bit. This bit
2			is set when an intervention required condition
3			occurs. It is reset when the intervention required
4			indicator is turned off.
5	2	7	Card removal timeout bit. This bit position is
6			reset to logic 0 at the beginning of each new user
7			transaction. The bit position is set to logic 1
8			whenever a predetermined period of time expires
9			following the availability of credit card to a
10			user without the card being removed from a terminal
11			14. This bit position indicates that some kind
12			of intervention is required. Normally, the host
13			data processing system would respond by commanding
14			the terminal to retain the credit card.
15	3	0	Open/close bit. This bit position is reset to logic
16			0 any time the terminal opens and is ready to
17			receive a user transaction request. This bit position
18			is set to logic 1 each time the terminal closes.
19	3	1	Cash out condition bit. This bit position is reset
20			at the beginning of each new user transaction. This
21			bit position is responsive to a hardware switch
22			which indicates whether or not there is enough cash
23			stored in the terminal to execute a maximum cash
24			issue transaction. The bit position is set to logic
25			1 any time the cash out condition occurs during the
26			execution of a preceding cash issue transaction to
27			which the status message corresponds. The setting
28			of this bit position indicates that intervention is
29			required and causes a terminal to close.

SA973016

# 1059630

1	3	2	Invalid backup encryption key bit. This bit position
2			is reset to logic 0 upon sending a status message and
3			is set to logic 1 upon receipt of a change key type
4			of command message from the host data processing
5			system containing an improper encryption key (an
6			improper encryption key contains all zeros).
7	3	3	Transaction statement dispenser form out bit. This
8			bit position is reset to logic 0 at the beginning
9			of each new user transaction. It is set to logic 1
10			when a transaction statement sensor indicates that
11			the last usable transaction statement form is
12			issued during the last preceding transaction to which
13			the status message corresponds.
14	3	4	Deposit flap (door) or issue gate open bit. This bit
15			position is reset upon sending a status message. The
16			bit position is set to logic 1 when the deposit flap
17			or issue gate remains open when it should be closed
18			and indicates that the flap or gate has been
19			jimmied.
20	3	5	Unrecoverable hardware failure bit. This bit position
21			is reset to logic 0 after an exception status
22			message has been set. This bit position is set to
23			logic 1 any time a jam or other error condition is
24			encountered which cannot be corrected, whether
25			during the execution of a transaction or at any
26			other time. Setting of this bit position indicates
27			that intervention is required and the terminal closes.
28	3	6	Customer door open bit. This bit position is reset
29			to logic 1 upon sending a status message. The bit
30			position is set to logic 1 when the customer door

## 1059630

1           which provides access to the user keyboard and  
2           display is open when it should be closed and  
3           indicates that the door has been jimmied. Setting  
4           of this bit indicates that intervention is required  
5           and causes the terminal to close.  
6        3     7       Security enclosure interlock bit. This bit position  
7           is reset to logic 0 when the operator access door  
8           is closed and set to logic 1 when the door is open.  
9           The terminal 14 closes any time this bit position  
10          is set to logic 1.

11           A transaction reply message from a host data processing  
12       system 12 to a user terminal 14 is generated in response to a user  
13       transaction request message. The transaction reply message begins  
14       with the standard four byte common header field specifying  
15       total message length (L), transaction number (N), message class  
16       (C), and message subclass (SC). Following the four bytes of the  
17       common header field are four bytes or 32 bits of encrypted  
18       information, a variable length optional display data field, a  
19       field separator character (FS) and a variable length optional  
20       transaction statement print field, and a final field separation  
21       character (FS). The four byte encrypted field includes a one  
22       byte cash counter 2 number (CNTR 2), a single action byte, a  
23       one byte cash counter 1 number (CNTR 1), and an amount byte (AMT)  
24       which specifies the number of bills for which the reply message  
25       is authorizing issuance. The terminal 14 checks this authorized  
26       amount against the request.

27           The action byte is a one byte instruction from the host  
28       data processing system 12 which directs a terminal 14 to con-  
29       summate a user transaction in a manner consistent with the data  
30       contents thereof.

## 1059630

1           Bit 0. When bit 0 is set to logic 1, a terminal 14  
2 is commanded to immediately display a standard terminal display  
3 message which is indicated by the optional display data field  
4 immediately following the encrypted field. Up to 128 separate  
5 messages designated 0-127 are stored in data storage 66 asso-  
6 ciated with the microprocessor 60. When bit 0 of the action  
7 byte is set to logic 1 the terminal 14 is commanded to display  
8 one of these messages which is indicated by the binary content  
9 of the one byte optional display field at byte position 9 of the  
10 transaction reply message.

11           Bit 1. When bit 1 is at logic one terminal 14 is  
12 commanded to immediately display an optional display message  
13 contained within the optional display data field immediately  
14 following the encrypted field. When bit 1 is set to logic one,  
15 byte 9 at the beginning of the optional display data field con-  
16 tains a binary number indicating the length of the display  
17 message in bytes exclusive of byte 9. Immediately following byte  
18 9 the transaction reply message contains the text of the desired  
19 display message in EBCDIC code with each byte indicating one  
20 display character.

21           Bit 2. A logic one at bit position two of the action  
22 byte indicates that a transaction terminal 14 is commanded to  
23 print information on a transaction statement and that the trans-  
24 action statement print data field of the reply message contains  
25 the data to be printed in EBCDIC code.

26           Bit 3 not defined.

27           Bit 4. A logic one in bit 4 indicates that a requested  
28 user transaction is authorized as requested.

29           Bit 5. A logic one in this bit position indicates that  
30 a user's credit card is to be retained by the terminal 14 while a



1 logic 0 indicates that the credit card is to be returned to  
2 the user.

3 Bit 6. A logic one in this bit position indicates that  
4 the user is required to acknowledge the transaction before the  
5 terminal 14 proceeds to execute the transaction. The user  
6 acknowledges the transaction by activating either a cancel key  
7 or a proceed key in a keyboard control field. Typically some  
8 indication of the transaction would be displayed at the time  
9 the user selects a key. For instance the message "TRANSFER  
10 \$50.00 FROM SAVINGS ACCOUNT TO CHECKING ACCOUNT - depress cancel  
11 or proceed" might be displayed.

12 Bit 7. Not defined.

13 The transaction statement print field at the end of  
14 a transaction reply message is divided into a plurality of  
15 subfields which permit the communication of print data for up to  
16 2 transaction statement forms. The first subfield is a common  
17 data subfield which carries information such as the user's name  
18 and account number which will be the same for both transaction  
19 statements. The common data field may either command a terminal  
20 14 to print a canned print message stored within the memory 66  
21 of a terminal 14 or may command the terminal to print a message  
22 transmitted as part of the common data field and standard EBCDIC  
23 code. The first byte of the common data field determines the  
24 source of the print data. If this byte contains a number from  
25 1 to 127 (below hex 80) the print data is contained in standard  
26 EBCDIC form in the common data subfield immediately subsequent  
27 to the first byte. In this instance, the first byte represents  
28 a binary length count indicating the number of bytes of text  
29 in the common data field exclusive of the length byte. If the  
30 common print data is to be provided by a canned message, a print

## 1059630

1 message ID number identifying the particular canned message is  
2 added to 128 (hex 80) and transmitted as the first and only byte  
3 of the common data subfield. By way of example, if the common  
4 data is to be taken from a canned message number 30, the one  
5 byte common data subfield would contain the binary number  
6  $30 + 128 = 158$  (hex 9E). A one byte data content corresponding  
7 to ID number 0 (hex 80) is used as a delimiter for the common  
8 data and statement data and must not be used to define message  
9  $\emptyset$  as a canned message. A statement number one data subfield  
10 immediately follows a delimiter byte hex 80 after the common  
11 data subfield. The statement number one data subfield may carry  
12 an actual EBCDIC print message or may identify a canned print  
13 message and uses the same format as the common data subfield.  
14 Print information commanded by the statement number one data  
15 subfield, however, will be printed only on one transaction state-  
16 ment form designated form one. The delimiter character (hex  
17 80) immediately follows the statement number one data subfield.  
18 A statement number two data subfield immediately follows the  
19 second delimiter character. The statement number two data sub-  
20 field has a format and data content similar to the common data  
21 subfield and statement number one data subfield. The statement  
22 number two data subfield may contain either a transmitted print  
23 message in EBCDIC code or identify a canned print message. If  
24 the statement number two data subfield is not present, i.e. has a  
25 length of 0 byte, a second transaction statement form is neither  
26 printed nor issued. A field separator character (FS) immediately  
27 follows the statement number two data subfield to indicate the end  
28 of the transaction statement print field and the end of a  
29 transaction reply message. Printing of a transaction statement  
30 form begins in the upper left hand corner and proceeds left to

## 1059630

1 right and line by line in the common English reading format.  
2 An EBCDIC carriage control code is utilized to terminate a line  
3 of text and begin the printing of the next textual character at  
4 the left most character position of the next line down. The  
5 printing operation follows a predetermined sequence in which  
6 common text is first printed on statement form one, statement  
7 one text is printed on statement form one, common text is printed  
8 on statement form two, and finally statement two text is printed  
9 on statement form two.

10 A command message is sent from the host data processing  
11 system 12 to a terminal 14 to control the operation or status of  
12 the terminal in accordance with the data content of the command  
13 message. Each command message begins with a four byte common  
14 header field and containing message length (L), transaction  
15 number (N), message class (C) and message subclass (SC). A four  
16 byte encrypted field follows the four byte header field. The  
17 four byte encrypted field includes the cash counter byte (CNTR1),  
18 the class and subclass byte (CNSC) containing both the class and  
19 subclass indication combined into a single byte, a second cash  
20 counter byte (CNTR2), and a special byte (SPEC). The special  
21 byte is utilized for an inquiry type of command message to  
22 indicate the information which is to be supplied by a responsive  
23 status message from a commanded terminal to the host data  
24 processing system. Bits 0-4 of the special byte are unassigned  
25 and are normally transmitted as logic 0. Bit 5 is set to logic  
26 one to indicate that a terminal is being commanded to retransmit  
27 its last status message. Bit 6 is set to logic one to indicate  
28 that the terminal is to transmit a current status message plus  
29 the 112 bytes of auxiliary storage within operator function  
30 subsystem 76 which do not contain the two encryption keys. A

1 logic one in bit 7 of the special byte indicates that the  
2 terminal is commanded to transmit a normal status message. Bits  
3 5, 6 and 7 are mutually exclusive where only one should be on  
4 at a time.

5 Two optional encrypted fields follow the common header  
6 field and four byte encrypted field of a command message. The  
7 first optional encrypted field carries a first half of an eight  
8 byte encryption key and the second optional encrypted field  
9 carries the second half of an eight byte encryption key. These  
10 first and second optional encrypted fields are included only  
11 following a set key or change key command. A terminal 14 responds  
12 to a change key command by deencrypting the command message with  
13 the old third or transmission encryption key (key B) and then  
14 substituting the key received in the optional encrypted fields  
15 one and two for all future communications. A set key command  
16 operates like a change key command except that the new key is  
17 encrypted in a backup key (key C) stored in the auxiliary memory.  
18 In a "change display message" type of command message the two  
19 optional encrypted fields are not included in the message but a  
20 clear text optional data field follows the four byte encrypted  
21 field. The clear text optional data field begins with an  
22 index number (INDX) followed by a data field length byte (LD) and  
23 new display text in standard EBCDIC code. A "change display mes-  
24 sage" type of command message does not affect the actual dis-  
25 play which is visible by a terminal user, but instead modifies  
26 the data content of a canned display message stored within the  
27 data storage 66. For example it may be desirable to change  
28 a canned display message "take out credit card" having a display  
29 message ID number 40 to "remove credit card". The index byte  
30 (INDX) contains the display message ID number of the canned

## 1059630

1 message which is to be changed. The data field length byte  
2 (LD) contains a binary number indicating the number of bytes  
3 in the text of the new message which immediately follows. If  
4 the new message is too long to fit into the number of bytes  
5 available in the table of display messages within data storage  
6 66, the command is not executed and the following status  
7 message indicates that the command was not executed. Because  
8 the display messages are of a variable length and because it is  
9 necessary for all messages from the host data processing system  
10 to a terminal 14 to contain an even number of bytes, it may  
11 be necessary to pad the end of a display text with an arbitrary  
12 pad character. This pad character would not be counted for  
13 the data field length byte (LD) but would be counted for the  
14 overall message length byte (L) in the common header field of  
15 the command message.

16           The load initialization message provides the  
17 information for the random access memory portion of data  
18 storage 66 which may have been lost in the event of a power  
19 shut down. It may also be used to reinitialize the terminal  
20 with new options. This message begins with the standard four  
21 byte common header field, followed by a two byte binary number  
22 field specifying the number of bytes in the following data  
23 field. The data field comprises the last field of the load  
24 initialization message and contains the customization image  
25 which is stored in data storage 66. The critical information  
26 such as micro program routines and option selection bytes  
27 in the data field is encrypted with the third transmission  
28 key (key B) in four byte sequential segments.

29           In general, the customization image which is received  
30 during initialization provides the information which may vary

1 from one terminal to another and is therefore not readily  
2 implemented with read only memories. Included within the  
3 customization image are the canned user display and print  
4 messages which may include up to 49 predetermined messages  
5 designated message 1-49. Also included as message 50 is an  
6 optional font table containing up to 574 bytes which permits the  
7 display of non-standard characters or graphics which have been  
8 custom selected by a given terminal customer such as a bank.  
9 Also included in the customization image is a certain amount  
10 of programming and program control information to account for the  
11 particular combination of available options which is implemented  
12 with a given terminal.

#### 13 TRANSACTION MESSAGE ASSEMBLY

14 The communications which are involved between a host  
15 data processing system 12 and a user transaction terminal 14  
16 during the execution of a requested user transaction are illus-  
17 trated in further detail in the operational block diagrams of  
18 Figs. 3-5 to which reference is now made. In order to facilitate  
19 an understanding of the operation of the invention, the operative  
20 communication system will be described in the context of specific  
21 user transaction examples. It should be appreciated however,  
22 that a transaction terminal 14 may perform any one of a large  
23 variety of user requested transactions and is not limited to  
24 these specific examples.

25 For a specific example it will be assumed that the  
26 terminal 14 is a through the wall terminal providing a walk up  
27 station at a branch bank. The through the wall terminal will be  
28 assumed to be connected in a manner similar to terminal 46 (Fig. 1)  
29 in a close loop to a controller 32 and through the controller 32  
30 to a host data processing system 12. The terminal 46 extends

1 through an exterior wall of the branch bank with the user  
2 communication facilities outside the bank and the majority  
3 of the terminal inside the bank. The operator maintenance  
4 access panel is accessible via the service door from the interior  
5 of the branch bank. As a potential user approaches the terminal  
6 46 the illumination of the keyboard area and a sign on the face  
7 of the terminal indicates that the terminal is in an available  
8 (open) condition. No light and a "closed" display indicate that  
9 the terminal is unavailable for the execution of transaction  
10 if the terminal is in a closed condition and any user action is  
11 ignored. If the terminal indicates an open condition, the pro-  
12 spective user initiates a user transaction by inserting his  
13 credit card into a slot. In this example, it will be assumed  
14 that a user desires to transfer funds from his savings account  
15 to his checking account.

16 1. TRANSACTION REQUEST MESSAGE

17 The first portion of the three part user transaction  
18 communication sequence is illustrated in Fig. 3. The terminal  
19 microprocessor 60 is shown only generally in Fig. 3 with no  
20 specific connections being made to physical or functional  
21 blocks. It will be appreciated that logical interconnection  
22 are as shown in Fig. 2 and that operational control and data  
23 processing are performed by the program microprocessor 60.

24 At the time the prospective user is issued a credit  
25 card 100 by the customer bank, he is also assigned a six digit  
26 personal identification (ID) number. This personal ID number  
27 may optionally be related to information recorded on a stripe  
28 of magnetic material on the credit card 100. As the card 100 is  
29 inserted into the terminal 46 the presence of the card is sensed  
30 and a credit card transport mechanism draws the card into the

## 1059630

1 terminal 14 and past a read head where the card is sensed for  
2 proper orientation and status. If the card is improperly  
3 oriented contains unreadable data, or of a type which cannot  
4 be accepted by the terminal 46 it is returned. (If the card  
5 is expired it may be retained upon host command) Assuming a  
6 proper credit card, the card 100 is transported past a card  
7 reader 102 where the information on the magnetic stripe is  
8 read and stored in the random of access portion of data  
9 storage 66 and the card is detained at a card escrow holding  
10 area. The credit card 100 is compatible with standards set  
11 forth by the American Bankers Association. This means that the  
12 magnetic stripe contains a sequence of five bit words representing  
13 a parity bit and four data bits. The four data bits include a  
14 start of card (SOC) character, a field separator character and an  
15 end of card (EOC) character. Numerals are indicated in binary  
16 coded decimal representation. A typical magnetic stripe format  
17 begins with a start of card (SOC) character followed by an  
18 account number of up to 19 characters, a field separator character,  
19 four characters specifying a month and year of the credit card  
20 expiration date, a discretionary data field, an end of card (EOC)  
21 character and a longitudinal redundancy check character. A  
22 maximum of 40-5 bit characters may be recorded on the magnetic  
23 stripe. As the characters are read a selection key designated  
24 k1 which is provided as an initialization option determines a  
25 starting point for selecting 8 sequential characters from the  
26 magnetic stripe. For example, if k1 contains the number 5, the  
27 fifth through 13 characters following SOM are selected at step 104  
28 without their parity bits to form 32 bits. These 32 bits are  
29 processed in an encryption algorithm 106 to generate 32 bits of  
30 encrypted data.



## 1059630

1           The comparison of part or all of a personal ID number  
2 with corresponding credit card information may be selectively  
3 provided as a customer option which is indicated at the time  
4 of initialization. If the comparison option is not selected  
5 the correspondence between ID numbers and credit card information  
6 may be randomly selected. However, the execution of a corres-  
7 pondence comparison is then impossible if the terminal 14 operates  
8 under control of an off line host. If the local check option is  
9 selected, two keys indicate the manner in which the check is  
10 executed.

11           The first check key, k1, permits the selection of  
12 any contiguous group of 8 characters read from the credit card.  
13 Key k1 identifies the position following SOM of the first of  
14 the 8 characters. The 8 characters would typically, but not  
15 necessarily, be chosen to be entirely within the credit card  
16 account number field. In the present example K1 = 5 causing  
17 characters 5-13 to be selected.

18           The second check key, K2, determines which digits  
19 within the personal ID number are to be checked by indicating  
20 the digit position at which the check is to begin. Thus, k2 = 1  
21 would cause digits 1-6 to be checked, k2 = 4 would cause digits  
22 4-6 to be checked and k2 = 6 would cause only the least signi-  
23 ficant digit to be checked as the number of checked digits  
24 increases (i.e. k2 smaller), the protection fraud by guessing  
25 at ID numbers is increased for off line host operation. However,  
26 the locally checked digits must have a predetermined correspondence  
27 with credit card information while non-checked digits may have  
28 a random correspondence. Increasing the number of locally  
29 checked digits thus decreases the number of digits available  
30 for random correspondence and increases the opportunity for

1059630

1 access to the data base of an on line host in the event that the  
2 correspondence algorithm and encryption key becomes compromised.  
3 For the present example it is assumed that the customer has  
4 exercised his option by selecting the local check feature  
5 with  $k_2 = 4$ .

6 The particular encryption algorithm which determines  
7 the correspondence between ID numbers and credit card information  
8 is not critical to the practice of this invention except that the  
9 relationship between the clear text input and encrypted text  
10 output should be dependent upon an encryption key designated  
11 here the first encryption key, Key A. For the purpose of this  
12 example it will be assumed that the encryption algorithm is of  
13 the type designated Lucifer in an article, H. Feistel, "Cryptography  
14 and Computer Privacy", Scientific American, May 1973, pp. 15-23  
15 or in an article, C. H. Meyer, "Enciphering Data for Secure  
16 Transmission," Computer Design, April 1974, pp. 129-134.

17 An encryption key such as Key A, for the algorithm 106, is  
18 a word containing 64 binary digits. The encryption key can also  
19 be thought of as including 8-8 bit bytes. Key A is stored  
20 within the auxiliary memory portion of operator function sub-  
21 system 76 and occupies 8 of the 128 memory words therein. In  
22 order to provide complete protection for this key, the key is  
23 destroyed each time a maintenance function from the customer  
24 interface panel is requested. This destruction prevents an  
25 ordinary terminal maintenance person from gaining access to the  
26 code. In one arrangement a trusted bank employee having access  
27 to Key A waits until the maintenance person completes terminal  
28 maintenance and then enters the 64 bit code as 8 sequentially  
29 entered hexadecimal digit pairs. An operator panel hexadecimal  
30 display indicates entered digits to permit correction if necessary

1 with only the two most recently entered digits being displayed  
2 at any given time. This restriction of the display to two digits  
3 protects the security of the key by requiring a person trying to  
4 copy the key by observation of the display to observe the  
5 display for a considerable period of time by making it impossible  
6 to observe the entire key at one instant as the key entry is  
7 completed. Once the key is entered it cannot be again displayed.  
8 It is thus possible to directly enter key A into the terminal as  
9 described above.

10           However, in an alternate example the trusted bank  
11 employee is given not Key A, but a Key A' having a predetermined  
12 relationship to Key A. In this example the trusted employee  
13 enters Key A' into the terminal in the same manner as if he  
14 were entering Key A itself. However, the terminal processes  
15 Key A' with an encryption algorithm 108 which may be similar  
16 to or even identical to encryption algorithm 106 to produce the  
17 encryption Key A. The encryption algorithm 108 uses a second  
18 encryption key designated Key C which is a terminal back up  
19 key in the encryption process which converts Key A' to Key A.  
20 Alternatively, a completely separate key could be loaded at  
21 initialization for this purpose.

22           Because of the predetermined relationship between the  
23 32 bits of credit card data which is encrypted with Key A and  
24 the 6 digit personal ID numbers which are given to a person at a  
25 time a card is issued, the security of Key A is extremely  
26 important. If a class of credit cards is to be usable at more  
27 than one branch of a customer bank, then at least one person at  
28 each bank must have access to Key A so that it can be keyed into  
29 a terminal 46 when necessary. For a large bank with many branches  
30 this distribution can become quite wide. Furthermore, if a

1059630

1 card is to be usable interchangeably at more than one bank, all banks accepting the card must have the same encryption Key A. The number of persons having access to Key A is thus further increased and can become quite substantial. The use of encrypted algorithm 108 provides security against this wide distribution of Key A. By using a difference Key C at each banking unit only a predetermined Key A' corresponding to the given Key C will operate satisfactorily to produce the highly important Key A. For example each unit might be a separate branch bank having three or four of the terminals 14. 10 Only the Key A' for that unit or branch bank will satisfactorily produce the Key A. If a person having access to Key A' at one branch goes to a different branch, where a different Key C is employed in encryption algorithm 108, the Key A' from the first branch will not produce the Key A at the second branch. It is thus possible to limit the distribution of Key A to a very small, highly select group of people.

20 Encryption algorithm 106 thus produces as an output 32 binary digits having a predetermined relationship to the 32 bit input. These 32 output bits are divided into 6-5 bit words in a table conversion process 110 with only 30 bits being used. For instance, the words may be formed from first 6 groups of five sequential bits each with the 1st two bits not being used. Each group of five bits is utilized in table conversion process 110 as an address word in accessing a table storing one decimal digit of value 1-9 at each address location. The table conversion thus results in 6 digits, each having a value of 1-9. These digits have a direct correspondence to the personal ID number and the digit 0 is excluded in order to avoid personal ID numbers which starts with leading 0's and can be expected to create confusion or variable length entries.

## 1059630

1           If the information on the credit card is found to be  
2 in order, a user access panel is opened to provide user access  
3 to the optical user display and user keyboard 112. The user is  
4 directed to enter his personal ID number through the numeric  
5 field of the keyboard. If the user does not enter exactly six  
6 digits within a predetermined period of time, an incorrect ID  
7 number is assumed and a retry is suggested. Upon entry of  
8 exactly six digits, a portion or the whole of the entered ID  
9 number is optionally compared with the 6 digit number generated  
10 by table conversion 110. The key K2 indicates which of the six  
11 corresponding pairs of digits are to be compared.

12           In this example it has been assumed that  $K2 = 4$  so  
13 that the three least significant digits having positions 4, 5  
14 and 6 are compared by compare step 114. If the comparison is  
15 invalid, a faulty ID number is indicated and the user is invited  
16 to retry entry of the ID number. If the ID number is not properly  
17 entered in a given number of retries such as 3, the transaction  
18 request is terminated and a message is sent to the host. Upon  
19 host command the credit card is preferably transported to a re-  
20 tention bin to prevent further use of the credit card in random  
21 attempts to match an ID number with a possibly stolen credit  
22 card. Alternatively, a credit card may be returned to the user.  
23 Upon determination that the compared digits of the keyed ID  
24 number match the corresponding digits which were obtained  
25 from the credit card, the six digits of the personal ID number  
26 are converted to a 32 bit binary code at step 116. In step 116  
27 the first 24 bits are obtained directly from the 6 entered digits.  
28 The last 8 bits or 1 byte is obtained by treating each sequential  
29 pair of four bit digits as a single byte and taking the successive  
30 "exclusive or" of corresponding bit positions in each of the

## 1059630

1 resulting three bytes to obtain the data content of the corres-  
2 ponding bit position in the fourth byte. Other means of obtaining  
3 the last 8 bits of information are acceptable so long as the  
4 method results in variable information which is a function of  
5 all bits of the entered ID number. These 32 bits are then pro-  
6 cessed with an encryption algorithm 118 using Key A to produce  
7 a 32 bit encrypted personal ID number. The encryption algorithm  
8 118 may in general be any suitable encryption algorithm, but for  
9 this example it will be presumed that it is identical to the  
10 encryption algorithm 106. Use of the same algorithm for both  
11 encryption processes permit use of the same stored program  
12 or hardware logic for both processes. The encryption key for  
13 algorithm 118 may also be in general any suitable key. However,  
14 for this example it is assumed that algorithm 118 utilizes Key A  
15 which is identical to the Key A utilized for algorithm 106.  
16 This multiple use of the same encryption key as well as the same  
17 encryption algorithm further reduces the complexity of the  
18 terminal 14 operation and the size of the required data storage.  
19 The 32 bits which result from encryption algorithm 118 thus  
20 represent an once encrypted personal ID number.

21         The 32 bits of the encrypted personal ID number are  
22 then converted in step 120 to 6 four bit digits with two four  
23 bit digits being dropped. In step 122 the two discarded digits  
24 are replaced by two four bit digits of variable data. This  
25 replacement of ID number derived information with variable  
26 information prevents the encrypted field from being a constant.  
27 In general the variable data may be any data which has no  
28 predetermined relationship to the personal ID number and which  
29 varies with each transaction request message. In this preferred  
30 embodiment, the variable data is a cash counter (CNTR) count for

1 cash issue transactions and a transaction number (N) for other  
2 transactions.

3         The 32 bits which result from the combination of the  
4 six four bit digits and the 8 bits of variable data are then  
5 passed through an encryption algorithm 124 which utilizes a third  
6 encryption Key B. Encryption algorithm 124 may in general be  
7 any suitable encryption algorithm. But for this preferred  
8 embodiment it will be assumed that algorithm 124 is identical  
9 to algorithm 118, algorithm 106, and algorithm 108. Key B is  
10 a 64 bit encryption key which is received from the host data  
11 processing system 12 during initialization and which cannot be  
12 changed except by communication of a new key from the host data  
13 processing system. The encryption algorithm 124 results in 32  
14 bits of encrypted data which are assembled in a transaction  
15 request message immediately behind the four byte common header  
16 as described previously.

17         After the compare step 114 at least partially validates  
18 the credit card, the user is instructed to indicate the trans-  
19 action which he is requesting by use of the keyboard 112. The  
20 user is first instructed to indicate the type of transaction  
21 which is being requested and all of the back lights in the  
22 transaction request field of the keyboard are illuminated.  
23 As a particular key, which in this case would be the funds  
24 transfer key, is activated, the back light of the activated key  
25 remains illuminated while the back lights of all other keys in  
26 the field are extinguished. The user is then instructed to  
27 select the account from which funds are to be transferred and  
28 the back lights of all of the keys in the from account field  
29 are illuminated. As the user selects the from savings key the  
30 back light of that key remains illuminated while the back lights

1059630

1 of all other keys in the from account field are extinguished.  
2 The user is then instructed to select the account to which the  
3 funds are to be transferred and all back lights in the to  
4 account field are illuminated. Upon selection of the checking  
5 account key, the activated key remains back lighted and the  
6 back lights of all other keys in the to account field are  
7 extinguished. The remaining back lights provide an audit trail  
8 so that a user may confirm or remind himself of the status of  
9 his transaction request entry. He can change his mind at any  
10 time by returning to a previously entered field activating a  
11 new key and continuing the keyboard entry process from that  
12 point. Numerical information such as the dollar amount of funds  
13 which are to be transferred is entered through the numeric field  
14 of keyboard 112. All entered numeric information is displayed  
15 for confirmation except the personal ID number. This number is  
16 not displayed in order to prevent surreptitious knowledge of the  
17 personal ID number by a person standing behind the user. The  
18 keyboard data, credit card data read from the magnetic stripe,  
19 and any desired additional data are then provided in clear text  
20 behind the four byte common header field and four byte encrypted  
21 field. This information is then communicated to the host data  
22 processing system 14 as a transaction request message.

23 2. TRANSACTION REPLY MESSAGE

24 Referring now to Fig. 4 as a transaction request message  
25 is received by the host data processing system 12, it undergoes  
26 processing 140 to separate the various fields of data with the  
27 common header field being used for message routing and with the  
28 32 encrypted bits being passed through a decrypt algorithm 142  
29 and the clear text being received by the host data processor 144  
30 which has a large data storage 146. The decrypt algorithm 142



1059630

1 uses Key B which is the same third or transmission key which was  
2 utilized for encryption algorithm 124. The host data processor  
3 12 utilizes the clear text data to access the user's data base  
4 record (file) data storage 146. This file contains account  
5 data as well as information associated with the user's credit  
6 card such as the encrypted personal ID number (or numbers).

7 The 32 bits which are generated by decryption  
8 algorithm 142 are passed through a separation processor 144  
9 wherein the 6 four bit digits of the encrypted personal ID number  
10 are separated from the two variable digits. A comparison 148  
11 is then performed with the communicated 6 digits of the encrypted  
12 ID number being compared with the 6 digits of ID information from  
13 the file which were stored in encrypted form.

14 This encryption process greatly improves the security  
15 of cash stored in the various transaction terminals 14 which  
16 may be in communication with an on line host data processing  
17 system. A person of ill intent who is in possession of the  
18 correspondence between credit card account numbers and personal  
19 ID numbers could surreptitiously obtain cash from the terminal 14.  
20 For example, a person might forge or steal credit cards having  
21 information stored thereon which pertains to actual user accounts.  
22 Using the forged credit card and the corresponding personal ID  
23 number, a person could first inquire as to the balance of various  
24 savings, checking or other accounts which are accessible through  
25 the credit card. Having obtained the balance information, the  
26 person could then use the credit card and the cash issue terminal  
27 14 to withdraw cash from those accounts until either the accounts  
28 or the terminal cash are depleted. Additional accounts with  
29 their credit card and corresponding personal ID number could be  
30 utilized in a similar manner until all of the cash at a cash

## 1059630

1 issue terminal has been issued. The person could then move on  
2 to deplete the cash from additional cash issue terminals in the  
3 system using additional credit cards and personal ID numbers.  
4 Because each cash terminal 14 may contain many thousands of dollars  
5 and because there may be many terminals 14 in communication with  
6 the host data processing system 12, it becomes extremely important  
7 to maintain the correspondence between credit card account numbers  
8 and personal ID numbers secured and yet permit local checks to  
9 allow higher availability of terminals 14 through off line use.  
10 It becomes extremely difficult for a person to obtain the corres-  
11 pondence between the credit card information and personal ID  
12 numbers for a large number of accounts when the techniques  
13 described herein are employed. Even if the personal ID number  
14 may be completely generated by passage of the stored credit card  
15 information through encryption algorithm 106, security of its  
16 encryption Key A is maintained as described above.

17         If the relationship of a portion (or preferably all),  
18 for example the first three digits, of the personal ID number and  
19 the stored credit card information has no predetermined relation,  
20 it becomes even more difficult to compromise the system. It is  
21 possible that personnel at the data processing center for the  
22 host data processing system 12 may have access to the stored  
23 encrypted ID number. However, the actual personal ID number is  
24 not stored in the host data processing system and the encrypted  
25 ID number is of no value in obtaining cash from a terminal 14  
26 since it is the actual personal ID number that must be entered  
27 through the keyboard of a terminal 14. It is thus necessary for  
28 a person seeking to obtain the correspondence between a large  
29 number of credit cards and corresponding personal ID numbers to  
30 have access to both the encrypted personal ID numbers stored in

1059630

1 the host data base and the decryption algorithm corresponding  
2 to encryption algorithm 118 and encryption Key A.

3 As credit cards are issued it is possible to limit the  
4 knowledge of the correspondence between credit card data and  
5 personal ID numbers to a very few people. In fact, accounts  
6 can be established with part of the personal ID number being  
7 derived from the credit card information, and part being  
8 randomly generated by a computer. The total personal ID number  
9 can then be printed and sealed in an envelope along with a  
10 credit card such that the personal ID number is available to  
11 human eyes only after the envelope is given to a prospective  
12 user as he opens a credit card account which can be processed  
13 by a terminal 14. It is thus possible to develop an assignment  
14 system wherein no banking personnel have access to the corres-  
15 pondence between credit card accounts and the associated personal  
16 ID number.

17 If the comparison 150 shows that the stored and communi-  
18 cated encrypted personal ID numbers are not identical, the host  
19 data processing system assembles and communicates a transaction  
20 reply message indicating that execution of the transaction is  
21 not authorized. The transaction reply message might direct  
22 the requesting terminal 14 to either retain or return the user  
23 credit card. On the other hand, if the stored and communicated  
24 encrypted ID numbers are found to correspond, and if the requested  
25 transaction does not violate any predetermined rules which might  
26 relate to dollar amounts, rates of withdraw, or account balances,  
27 the transaction is authorized by a transaction reply message.  
28 The transaction reply message contains 32 bits of encrypted  
29 information corresponding to the 32 bits of encrypted inform-  
30 ation which are received in the transaction request message.

## 1059630

1 In an assembling process 152, 32 bits are assembled for encryption  
2 with encrypted algorithm 154 using Key B, which is the third,  
3 transmission encryption key. The encryption algorithm may in  
4 general be any suitable encryption algorithm but for this  
5 example it is assumed that the algorithm is identical to encryption  
6 algorithms 106, 118 and 124. It is further assumed that Key B  
7 is identical to the Key B for encryption algorithm 124. The 32  
8 bits which are assembled for encryption are different from the  
9 communicated 32 bits which contained the 6 digits of encrypted  
10 ID number and two variable digits. The 32 bits of the trans-  
11 action reply message include a one byte cash counter number  
12 corresponding to a first cash count (CNTR 1) maintained by a  
13 terminal 14 which is incremented for each bill issued, an action  
14 byte which indicates the response the terminal 14 is to take  
15 to the requested user transaction, a second cash counter byte  
16 (CNTR 2) identifying the cash count which is maintained for a  
17 second cash issue mechanism within the terminal 14 and an amount  
18 byte (AMT) which indicates the number of bills which is relevant  
19 to the requested transaction. These 32 bits are then processed  
20 through encryption algorithm 154 to form 32 encrypted bits 156.  
21 The encrypted bits 156 are then combined with clear text data such  
22 as optional display data, optional receipt data, or additional  
23 data required to complete the transaction and communicated back  
24 to the requesting terminal 46 and as a transaction reply message  
25 in step 158.

### 26 3. EXECUTION AND STATUS MESSAGE

27 As the transaction reply message is received by the  
28 terminal 14 the message undergoes input processing 160 to check  
29 for transmission accuracy and separate the reply message into  
30 its various fields. The encrypted field is passed through a

## 1059630

1 decryption algorithm 162 which uses Key B to restore the 32 bits  
2 containing the cash counter one (CNTR 1), ACTION, cash counter  
3 two (CNTR 2) and amount data (AMT). These bytes are checked  
4 for accuracy to ensure that the transaction reply message was  
5 received error free and that it corresponds to the correct  
6 transaction request message. A transaction conclusion 164  
7 is then executed in accordance with the contents of the trans-  
8 action reply message. In concluding a transaction, the terminal  
9 14 returns or retains the credit card, issues appropriate documents  
10 such as cash or printed transaction statements, formally executes  
11 or cancels the transaction, displays appropriate messages to allow  
12 the user's approval or disapproval; and performs any additional  
13 transaction execution functions which are necessary for completion  
14 of the transaction.

15       Upon completion of a user requested transaction, the  
16 terminal 14 communicates a status message to the host data  
17 processing system 12 which informs the data processing system 12  
18 of the manner in which the requested transaction was terminated  
19 and the status of the terminal 14. Preparation of the status  
20 message includes assembly 166 of 32 bits which are encrypted  
21 with encrypted algorithm 168 using Key B to generate 32 encrypted  
22 bits 170. The encryption algorithm 168 may in general be any  
23 suitable encryption algorithm but for the preferred embodiment  
24 presented herein, encryption algorithm 168 is identical to  
25 encryption algorithms 106, 108, 118, 124, and 154. Key B is  
26 identical to Key B for encrypted algorithms 124 and 152. However,  
27 unlike Key A, Key B may be changed by the host data processing  
28 system 12 and it is anticipated that Key B would be changed from  
29 time to time. The 32 bits 170 undergo output processing 172  
30 as they are combined with non-encrypted status information

1059630

1 and transmitted as a status message from the transaction ex-  
2 ecution terminal 14 to the host data processing system 12.

3           The method of using encryption algorithms as described  
4 herein provides great security for the transaction execution  
5 system 10 without requiring the storage capacity for storing the  
6 multiple encryption programs. Furthermore, with the proper selection  
7 of the encryption and decryption algorithms, the decryption  
8 algorithm can be quite similar to the encryption algorithm to  
9 permit a double usage of most of the encryption algorithm  
10 program for both encryption and decryption. This results in  
11 a further savings of program storage requirements. The last  
12 encryption of the 32 bits of encrypted information in the three  
13 user transaction messages permits security of the encrypted ID  
14 number along the communication channel while permitting the  
15 same general format to be utilized for all three messages. In  
16 the transaction request message, assembly process 122 combines the  
17 encrypted ID number with varying data to make it extremely  
18 difficult for a person monitoring the communication lines to break  
19 Key B and encryption algorithm 124 by repeatedly entering the  
20 same ID number, credit card and request and monitoring the  
21 corresponding encrypted communications. The transaction reply  
22 message contains a counter one byte, an action byte, a counter  
23 two byte and an amount. This information is all different from  
24 the encoded information of the transaction request message and also  
25 contains varying information. The amount and the action byte  
26 will tend to be the same for the same types of transaction request,  
27 however the control bytes will change. The 32 encrypted bits  
28 of the status message are different from the encrypted fields of  
29 either of the other two messages in that they contain the trans-  
30 action number which is time varying, the counter two and counter

## 1059630

1 one bytes in different byte positions from the transaction reply  
2 message and a count byte (CB) which indicates (via a binary count)  
3 the number of status and inquiry data bytes which follow the  
4 encrypted portion of the message for a normal status message.  
5 A status message which is generated in response to a transaction  
6 termination would normally contain no inquiry data byte. In the  
7 event that the status message is a "request recovery" type of  
8 exception status message, the third byte (CB) of the encryption  
9 field contains the "action" byte from the transaction reply  
10 message for the last request. Thus, by changing Key B from  
11 time to time and sending different information in the encrypted  
12 portion of each different type of message, the task of breaking  
13 the transmission encryption algorithm and finding the current  
14 Key B by monitoring the communication lines is made extremely  
15 difficult. Even if the transmission encryption algorithm and  
16 Key B were broken, monitoring the transmission of messages would  
17 produce a correspondence between accounts and encrypted personal  
18 ID numbers only for specific credit cards which are used while  
19 the communication is being monitored. The assembly of a large  
20 number of forged or stolen credit cards and corresponding personal  
21 ID numbers could be accomplished only by further breaking the  
22 Key A. In an alternative embodiment where there is a predetermined  
23 relationship between all digits of the personal ID number and  
24 information on the credit card, access to the data base is not  
25 necessary. Keys K1 and K2 of course provide further security  
26 for the encrypted ID number in the event that the local ID  
27 check option is implemented.

28 While there have been described above various arrange-  
29 ments of transaction execution systems in accordance with the  
30 invention for the purpose of illustrating the manner in which

**1059630**

- 1 the invention may be used to advantage, it will be appreciated
- 2 that the invention is not limited thereto. Accordingly, any
- 3 modification, variation or equivalent arrangement within the
- 4 scope of the accompanying claims should be considered to be
- 5 within this scope of invention.
- 6 WHAT IS CLAIMED IS



The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

- 1           1. A transaction terminal which is connectable to a  
2 host and which is dependent upon a host for approval and recor-  
3 dation of transactions indicated by a user, the transaction  
4 terminal comprising:
  - 5           a data input device for entering a user determined  
6 block of identification information;
  - 7           an encoder connected to encode at least a portion of  
8 the block of identification information to produce a first  
9 encrypted block of identification information indicative of  
10 at least a portion of the identification block of information;
  - 11          an encoder connected to encode at least a portion  
12 of the first encrypted block of identification information to  
13 produce a second encrypted block of identification information  
14 indicative of at least a portion of the identification block  
15 of information; and
  - 16          a transmitting system connected to transmit at least  
17 a portion of the second encrypted block of identification  
18 information to a host.

**1059630**

1           2. The transaction terminal as set forth in claim 1  
2     above, further comprising a device for generating a block of  
3     variable information which changes with each user transaction  
4     and wherein the second encrypted block producing encoder is  
5     further connected to encode a block of variable information  
6     along with the at least a portion of the first encrypted  
7     block of identification information to produce a second  
8     encrypted block of identification information indicative of  
9     both variable information and at least a portion of the  
10    block of identification information.

1           3. The transaction terminal as set forth in claim 1  
2     above, further comprising means for storing first and second  
3     encryption keys and wherein the first and second encoded blocks  
4     of information are produced in response to the first and second  
5     encryption keys respectively.

**1059630**

1           4. The transaction terminal as set forth in claim  
2   3 above, further comprising an operator control panel for  
3   entering operator determined information;  
4           means responsive to the entry of control information  
5   through the operator panel for displaying first encryption key  
6   anytime; and  
7           an encoder connected to produce and store the first  
8   encryption key in response to the entry of proper operator  
9   determined information through the control panel.

1           5. The transaction terminal as set forth in claim 4  
2   above, further comprising means for storing a third encryption  
3   key and wherein the first encryption key is produced in response  
4   to a stored third encryption key and information entered through  
5   the control panel.

1           6. The transaction terminal as set forth in claim 1  
2     above, wherein the block of identification information is of a  
3     length less than a predetermined length and further comprising  
4     means for expanding a data block length connected to receive a  
5     short data block of a length less than a predetermined length  
6     from the data input device, expand the received data block to a  
7     predetermined length by adding characters which are dependent  
8     upon the data content of the short data block, and provide a  
9     data block which has been expanded to a predetermined length  
10    to the first encrypted block producing encoder.

1           7. The transaction terminal as set forth in claim 6  
2     above, wherein the expanding means expands a short block of  
3     data by adding characters which are generated from the process  
4     of taking the logical exclusive-or of selected portions of  
5     the short block of data.

## 1059630

1           8. The transaction terminal as set forth in claim 1  
2     above, further comprising means for reading prerecorded infor-  
3     mation from a user produced card, an encoder connected to  
4     encode a selected portion of the prerecorded information read  
5     from a card to produce a block of encrypted card information,  
6     and a comparator connected to compare a selected portion of  
7     the block of identification information received by the data  
8     input device with a corresponding selected portion of the block  
9     of encrypted card information and indicate the identity or non-  
10    identity of the compared data.

1           9. The transaction terminal set forth in claim 8  
2     above, further comprising means responsive to the identity or  
3     nonidentity indication for inhibiting the transmission of said  
4     at least a portion of the second encrypted block to a host.

**1059630**

1           10. The transaction terminal as set forth in claim 8  
2 above, further comprising means for selecting a portion of the  
3 prerecorded card information for encoding in response to a  
4 predetermined encryption key.

1           11. The transaction terminal as set forth in claim 8  
2 above, further comprising means for selecting a portion of the  
3 block of identification information and block of encrypted  
4 card information which are to be compared in response to a  
5 predetermined encryption key.

1059630

1           12. A transaction execution system comprising:  
2           a host data processing system which is operable to  
3 maintain a plurality of accounts, approve or disapprove requested  
4 transactions affecting the maintained accounts, and modify main-  
5 tained accounts in accordance with approved requested transactions  
6 which affect said accounts, a plurality of said accounts each  
7 including a first block of information and a second block of  
8 information which is obtainable by encrypting a third block of  
9 information in accordance with a first predetermined encoding  
10 scheme, the host data processing system being operable to approve  
11 a requested transaction and correspondingly modify a related  
12 information block containing account only when both the first  
13 and second blocks of information for an account which is related  
14 to a requested transaction are included as part of a transaction  
15 request received by the host; and  
16           at least one transaction execution terminal in com-  
17 munication with the host data processing system, the terminal  
18 being operable to receive transaction request information from  
19 a user along with first and third blocks of information for an  
20 adversely affected account, the terminal including means for  
21 encrypting the third block of information in accordance with  
22 the first predetermined encoding scheme to obtain a second  
23 block of information and means for communicating the trans-  
24 action information, first block of information and second  
25 block of information to the host data processing system.

1059630

1           13. The transaction execution system as set forth in  
2     claim 12 above, wherein the transaction execution terminal  
3     includes means for issuing cash to a user in response to a cash  
4     issue transaction request which is approved by the host upon  
5     receipt of a host approval indication, and wherein the host data  
6     processing system is operable to communicate a cash issue trans-  
7     action approval indication to the terminal in response to the  
8     receipt from the terminal of cash issue transaction request  
9     information and first and second blocks of information if pre-  
10    determined conditions are met, said predetermined conditions  
11    including a predetermined correspondence between the first and  
12    second blocks of information received by the host data processing  
13    system as from the terminal.

1           14. The transaction execution system as set forth in  
2     claim 13 above, wherein the first block of information is  
3     received by the terminal by reading the information from a  
4     user supplied card and indicates a user account within the  
5     plurality of information block accounts.



1           15. The transaction execution system as set forth  
2 in claim 14 above, wherein the third block of information is  
3 obtainable by encrypting the second block of information in  
4 accordance with a second predetermined encoding scheme.

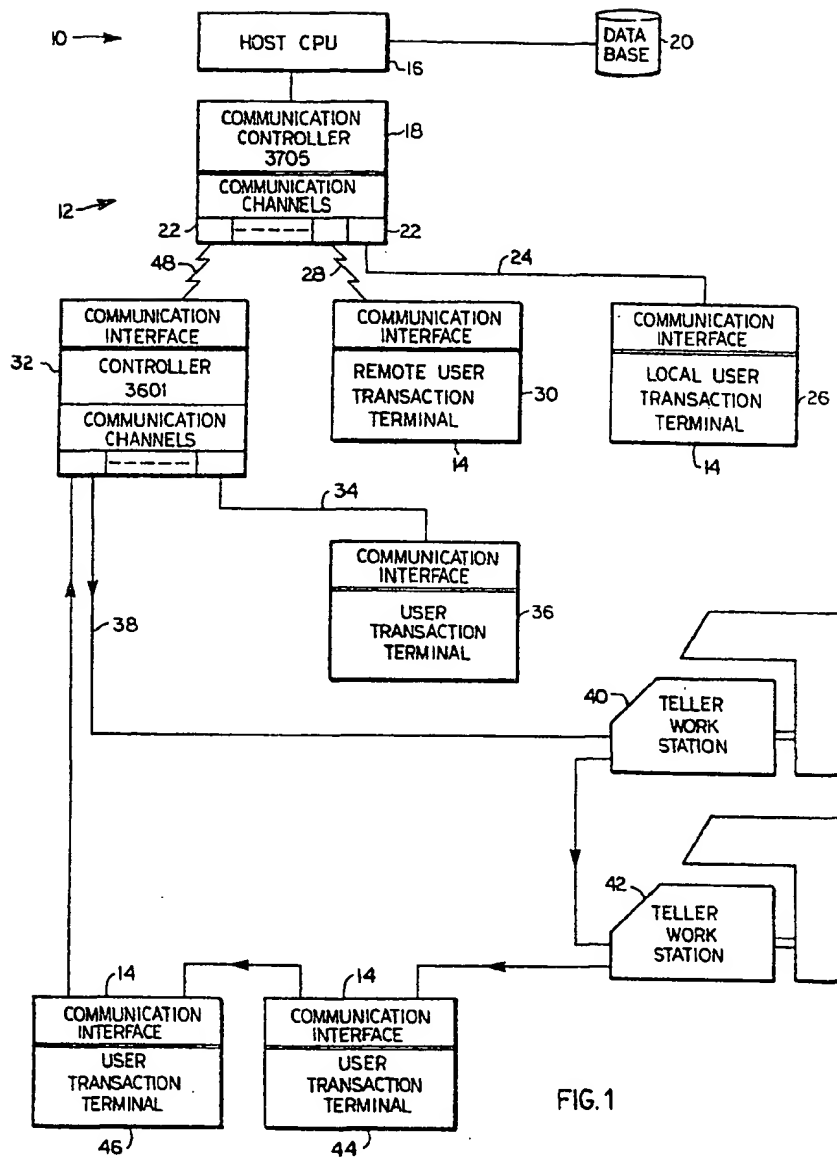
1           16. The transaction execution system as set forth  
2 in claim 15 above, wherein the first and second predetermined  
3 encoding schemes are the same.

1           17. The transaction execution system as set forth  
2 in claim 12 above, wherein the first block of information is  
3 received by the terminal by reading the information from a user  
4 supplied card.

1           18. The transaction execution system as set forth  
2 in claim 17 above, wherein the third block of information is  
3 obtainable by encrypting the second block of information in  
4 accordance with a second predetermined encoding scheme.

1           19. The transaction execution system as set forth  
2 in claim 18 above, wherein the first and second predetermined  
3 encoding schemes are the same.





*[Signature]*  
PATENT AGENT

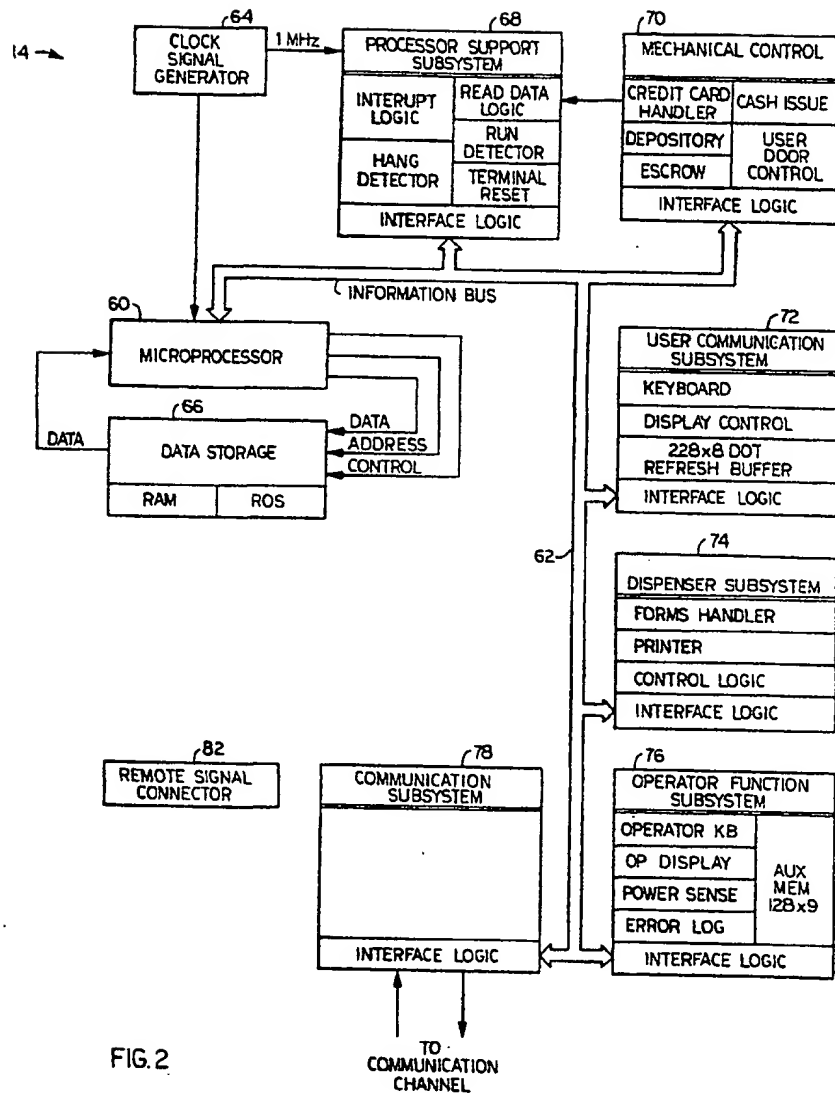
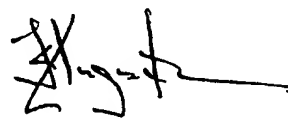
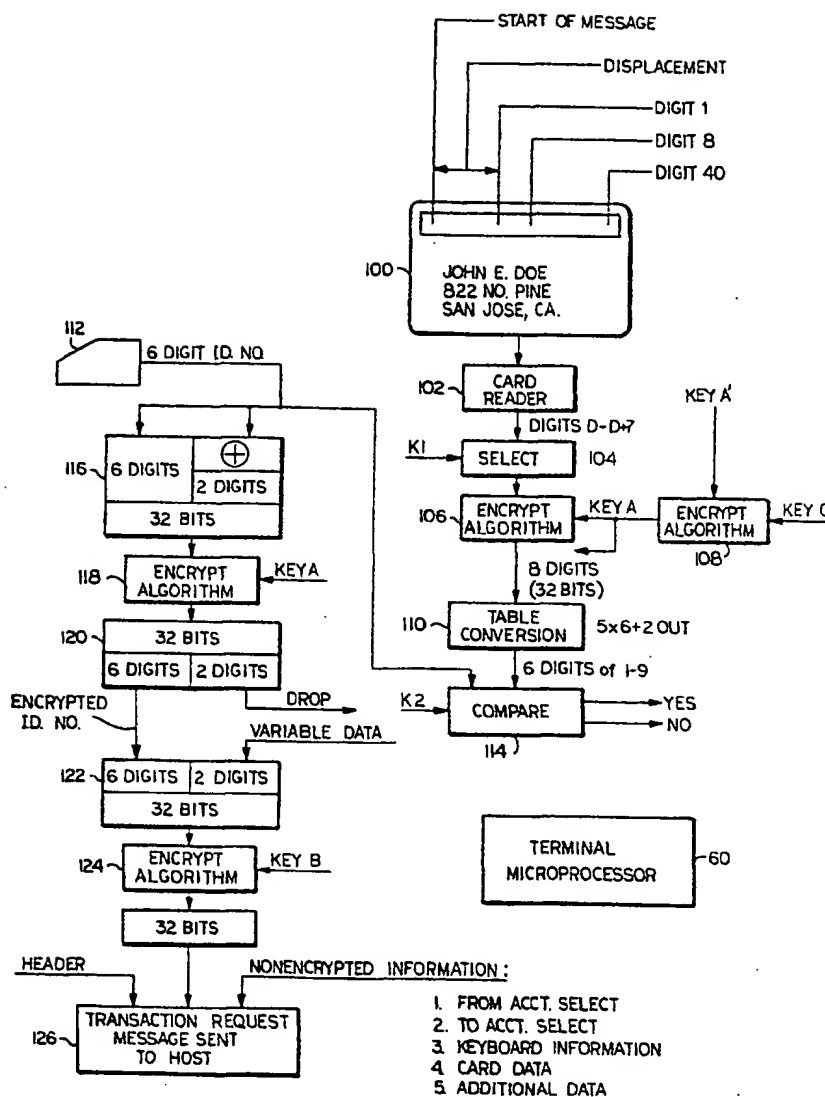


FIG. 2



PATENT AGENT



PATENT AGENT

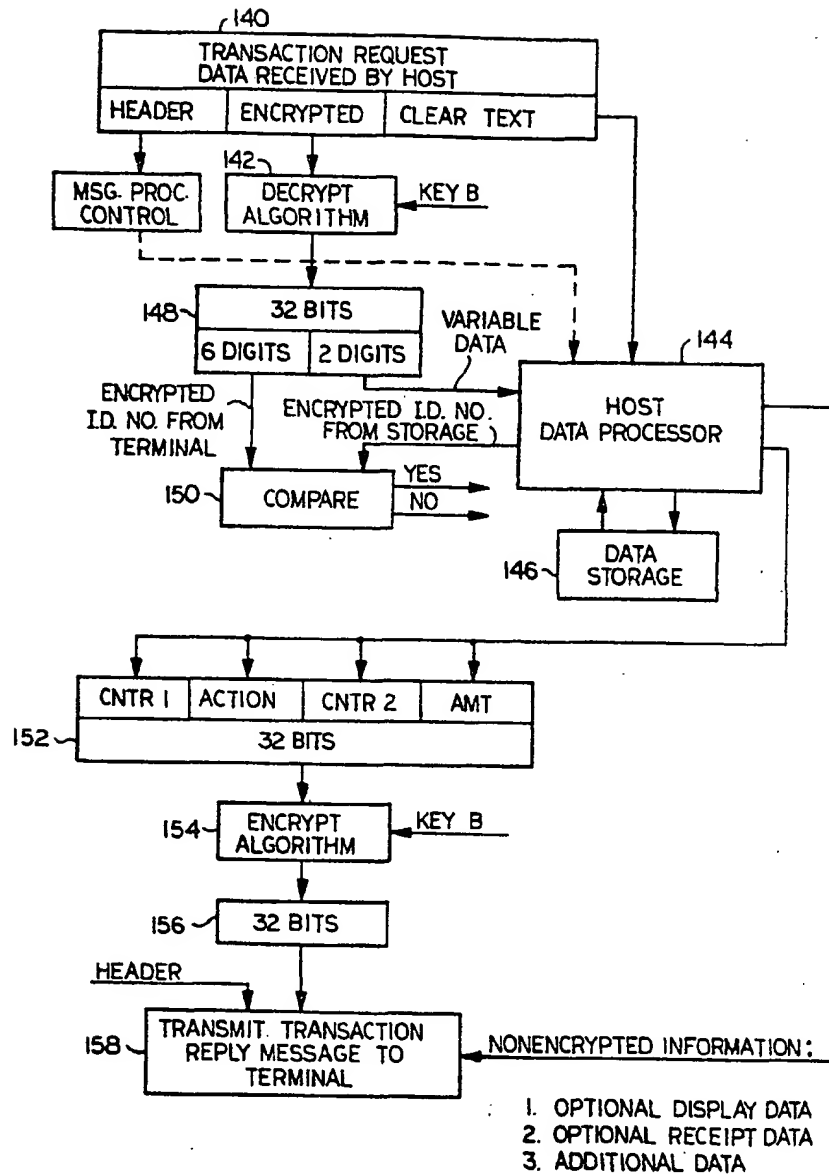
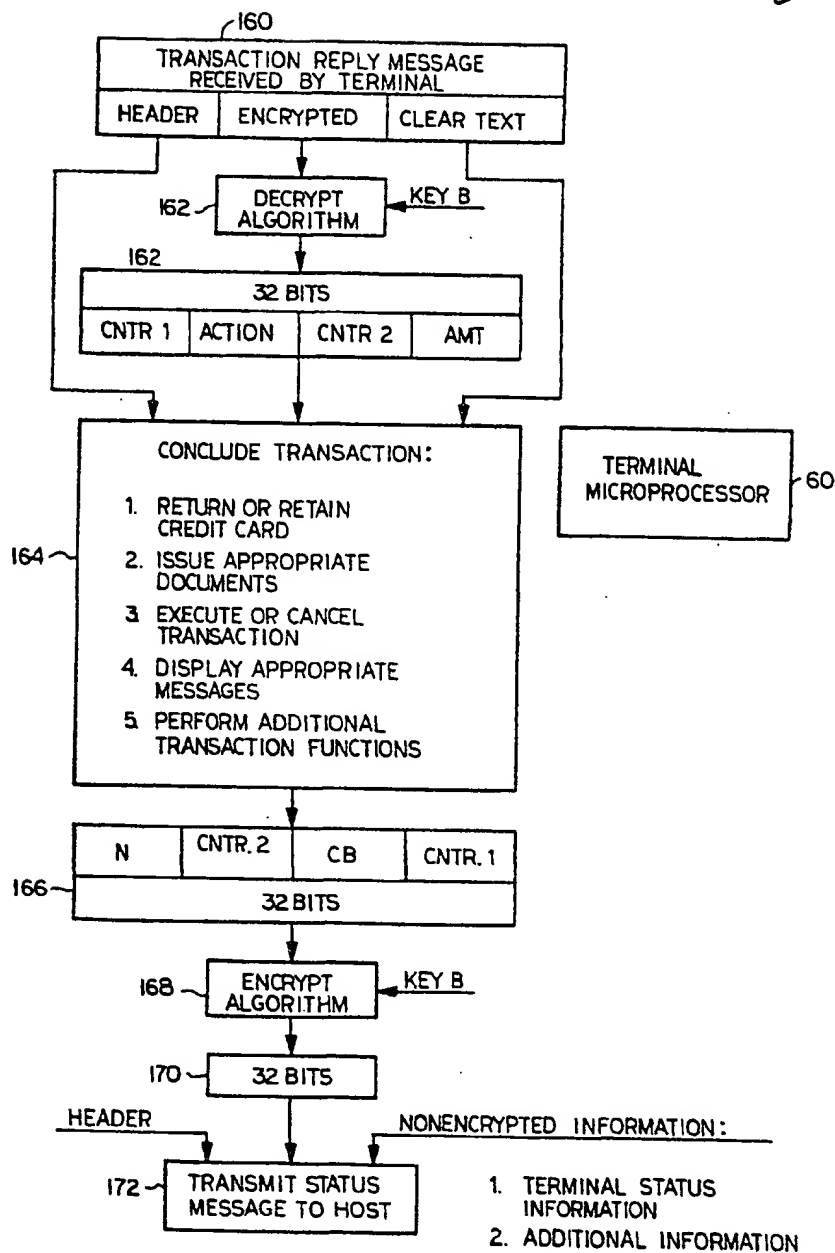


FIG. 4

PATENT AGENT



21-10-19

PATENT AGENT

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**